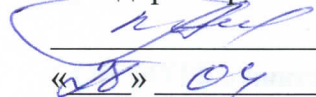


МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ
И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Северо-Кавказский филиал
ордена Трудового Красного Знамени федерального государственного
бюджетного образовательного учреждения высшего образования
«Московский технический университет связи и информатики»

Утверждаю

Зам. директора по УР



Н.А. Андреева

2026 г.

Разработка кроссплатформенных приложений Б1.В.ДВ.05.01 рабочая программа дисциплины

Кафедра
Направление подготовки
Профиль
Формы обучения

«Информатика и вычислительная техника»
09.03.01 Информатика и вычислительная техника
Искусственный интеллект и машинное обучение
очная, заочная

Распределение часов дисциплины по семестрам (ОФО), курсам для заочной формы (ЗФО))

Вид учебной работы	ОФО		ЗФО	
	ЗЕ	часов	ЗЕ	часов
Общая трудоемкость дисциплины, в том числе (по семестрам, курсам):	3	108/8	3	108/4
Контактная работа, в том числе (по семестрам, курсам):		48/8		18/4
Лекции		24/8		8/4
Лабораторных работ				
Практических занятий		24/8		10/4
Семинаров				
Самостоятельная работа		24/8		81/4
Контроль		36/8		9/4
Число контрольных работ (по курсам)				1/4
Число КР (по семестрам, курсам)				
Число КП (по семестрам, курсам)				
Число зачетов с разбивкой по семестрам				
Число экзаменов с разбивкой по семестрам (курсам)		1/8		1/4

Программу составил:
Доцент кафедры ИВТ к.т.н. доцент Лобзенко П.В.

Рабочая программа дисциплины
«Разработка кроссплатформенных приложений»

Разработана в соответствии с ФГОС ВО:
направления подготовки **09.03.01 Информатика и вычислительная техника**, утвержденным
приказом Министерства науки и высшего образования Российской Федерации от 19 сентября
2017 г. N 929

Составлена на основании учебных планов
направления **09.03.01 Информатика и вычислительная техника**
профиль "Искусственный интеллект и машинное обучение", одобренных Учёным советом
СКФ МТУСИ, протокол № 8 от 31.03. 2026г, и утвержденного директором СКФ
МТУСИ 31.03 2026 г.

Рассмотрена и одобрена на заседании кафедры
"Информатика и вычислительная техника"

Протокол от 28.04 2026 г. № 9

Зав. кафедрой  Соколов С.В.

1. Цели изучения дисциплины

Целями изучения дисциплины «Разработка кроссплатформенных приложений» являются:

- изучение алгоритмов численного решения инженерных задач, используемых в процессе эксплуатации компонентов информационных систем;
- приобретение студентами знаний и навыков практического использования различных приемов численного решения инженерных задач при разработке компонентов информационных систем и средств связи.

2. Планируемые результаты обучения

Изучение дисциплины направлено на формирование у обучающегося способности решать профессиональные задачи в соответствии с *проектной деятельностью*.

Результатом освоения дисциплины являются сформированные у обучающегося следующие компетенции:

Компетенции обучающегося, формируемые в результате освоения дисциплины (в части, обеспечиваемой дисциплиной)	
ПК-4 Способен разрабатывать программное обеспечение с применением методов машинного обучения	
Знать:	
- методы машинного обучения, приемы их оценивания и использования.	
Уметь:	
- осуществлять оценку и выбор алгоритмов машинного обучения в соответствии с поставленной задачей;	
- выполнять поиск, подготовку и разметку структурированных и неструктурированных данных для машинного обучения;	
- разрабатывать программное обеспечение на основе алгоритмов машинного обучения.	
Владеть:	
- технологиями разработки программного обеспечения с применением методов машинного обучения.	

3. Место дисциплины в структуре образовательной программы

Требования к предварительной подготовке обучающегося (предшествующие дисциплины, модули, темы):	
1	Б1.О.08 «Информационные технологии и программирование»
2	Б1.В.08 «Разработка полного цикла»
3	Б1.В.10 «Функциональное программирование»
Последующие дисциплины и практики, для которых освоение данной дисциплины необходимо:	
1	Б1.В.03 «Машинное обучение»
2	Б1.В.11 «Системы машинного зрения»

4. Структура и содержание дисциплины

4.1 Очная форма обучения, 4 года (всего 108 часов, 48 часов контактной работы)

Код зан.	Тема и краткое содержание занятия	Вид зан.	Кол. часов	Компетенции	УМИО
1	2	3	4	5	6
Курс 4, Семестр 7					
Модуль 1: Основы создания кроссплатформенных приложений - 36 (24+12) часов					

1.1	Введение. Назначение курса и особенности его освоения. <i>Структурные языки программирования в сравнении с другими видами языков высокого уровня. Управляющие операторы языка C++</i>	Лек.	2	ПК-4	Л1.1
1.2	Операторы управления в C++. <i>Составной оператор. Условные операторы. Операторы циклов.</i>	Лек.	2	ПК-4	Л1.1, Л2.1
1.3	Использование функций при программировании на C++. <i>Объявление функции. Передача параметров в функцию и возврат результата с помощью оператора return. Решение задач с использованием функций.</i>	Лек.	2	ПК-4	Л1.1, Л2.1
1.4	Массивы в C++. <i>Статические и динамические массивы. Основные алгоритмы обработки массивов. Указатели на функции. Статические и динамические матрицы и их обработка в C++.</i>	Лек.	2	ПК-4	Л1.1
1.5	Организация ввода-вывода в C++. <i>Форматированный ввод-вывод в C++. Работа с текстовыми файлами в C++. Функции fscanf() и fprintf().</i>	Лек.	2	ПК-4	Л1.1, Л2.1
1.6	Строки и структуры в языке C++. <i>Операции над строками. Тип данных string. Общие сведения о структурах.</i>	Лек.	2	ПК-4	Л1.1, Л2.1
1.7	Создание базовых приложений со структурами в C++. <i>Обработка массивов и матриц. Обработка строк в C++.</i>	ПЗ	4	ПК-4	Л1.1, Л2.1, Л3.2
1.8	Создание приложений с различными видами наследования. <i>Практическая реализация различных видов наследования классов.</i>	ПЗ	4	ПК-4	Л1.1, Л3.1
1.9	Составление приложений с инкапсуляцией и полиморфизмом. <i>Практическое создание скрытых методов и полей, а также полиморфных методов классов.</i>	ПЗ	4	ПК-4	Л1.1, Л3.1
1.10	<i>Простые и составные операторы. Использование многомерных массивов. Практическое изучение использования функций. Файлы и работа с ними. Структуры, составление и использование структур. Многозадачное программирование. Шаблоны классов. Элементы стандартной библиотеки C++. Практическое использование функций в C++. Функция main(). Рекурсивные функции. Перегрузка функций. Шаблоны функций. Разработка консольных приложений в C++. Обработка массивов и матриц. Создания различных структур в C++. Исследование приложений с различными видами наследования. Исследование различных видов наследования классов. Исследование приложений с инкапсуляцией и полиморфизмом методов классов. Исследование особенностей скрытых методов и полей, а также полиморфных методов классов.</i>	СРС	12	ПК-4	Л1.1, Л2.1, Л3.2

Модуль 2. Разработка кроссплатформенных приложений в среде Qt -36 (24+12) часов					
2.1	Знакомство с Qt. Подготовка к работе. <i>Справка и ресурсы. Обзор настроек среды Qt Creator. Создание первого проекта.</i>	Лек.	2	ПК-4	Л1.1, Л2.1
2.2	Структура проекта. Основные типы. <i>Компиляция проекта. Консольный проект Qt. Вывод сообщений. Работа с текстовыми строками в Qt.</i>	Лек.	2	ПК-4	Л1.1, Л2.1
2.3	Классы и файлы в Qt. <i>Контейнерные классы в Qt. Работа с файлами.</i>	Лек.	2	ПК-4	Л1.1, Л2.1
2.4	Создание консольных приложений в Qt. <i>Исследование особенностей создания различных кроссплатформенных приложений.</i>	ПЗ	4	ПК-4	Л1.1, Л3.2
2.5	Создание графического интерфейса средствами Qt. <i>Виджеты (Widgets). Компоновка (Layouts). Создание сигналов (signals) и слотов (slots).</i>	Лек.	2	ПК-4	Л1.1, Л2.1
2.6	Создание элементов графического интерфейса. <i>Класс QObject. События (Events). Обработка событий (Event handling). Создание собственного элемента интерфейса. Программирование формы созданной в Qt Designer.</i>	Лек.	2	ПК-4	Л1.1, Л2.1
2.7	Реализация приложений с интерфейсами пользователя в C++ . <i>Программная реализация оконных приложений в Qt. Решение практических задач.</i>	ПЗ	4	ПК-4	Л1.1, Л3.1
2.8	Приложения с элементами ООП. <i>Реализация ООП приложений в Qt.</i>	ПЗ	4	ПК-4	Л1.1, Л3.2
2.9	Заключение. Краткий обзор изученного материала. <i>Подведение итогов модульно-рейтинговой системы. Перспективы развития процедурных языков программирования.</i>	Лек.	2	ПК-4	Л1.1, Л2.1
2.10	<i>Изучение вариантов составления структур. Простые и управляющие операторы. Изучение команд препроцессора и функций пользователя. Файлы и работа с ними. Библиотеки для работы с комплексными числами. Обработка двоичных файлов. Перегрузка функций и конструкторов. Организация рекурсий в функциях.</i>	СРС	12	ПК-4	Л1.1, Л2.1, Л3.2
	Экзамен		36	ПК-4	Л1.1-Л3.2, Э1-Э5
Итого – 108 часов					

4.2 Заочная форма обучения 5 лет (всего 108 часов, 18 часов контактной работы)

Код зан.	Тема и краткое содержание занятия	Вид зан.	Кол. часов	Компетенции	УМИО
1	2	3	4	5	6
Курс 4					
Модуль 1: Основы создания кроссплатформенных приложений - 36 (6+30) часов					
1.1	Создание и удаление объектов. <i>Виды объектов. Работа с объектами. Доступ к объектам и время существования объектов.</i>	Лек.	2	ПК-4	Л1.1, Л1.2

1.2	Создание приложений с различными видами наследования. <i>Практическая реализация различных видов наследования классов.</i>	ПЗ	4	ПК-4	Л1.1, Л3.1
1.3	<i>Операторы управления в C++. Составной оператор. Условные операторы. Операторы циклов. Простые и составные операторы. Использование многомерных массивов. Практическое изучение использования функций. Файлы и работа с ними. Структуры, составление и использование структур. Многозадачное программирование. Шаблоны классов. Элементы стандартной библиотеки C++. Использование функций при программировании на C++. Объявление функции. Передача параметров в функцию и возврат результата с помощью оператора return. Решение задач с использованием функций. Массивы в C++. Статические и динамические массивы. Основные алгоритмы обработки массивов. Указатели на функции. Статические и динамические матрицы и их обработка в C++. Организация ввода-вывода в C++. Форматированный ввод-вывод в C++. Работа с текстовыми файлами в C++. Функции fscanf() и fprintf(). Строки и структуры в языке C++. Операции над строками. Тип данных string. Общие сведения о структурах. Практическое использование функций в C++. Функция main(). Рекурсивные функции. Перегрузка функций. Шаблоны функций. Разработка консольных приложений в C++. Обработка массивов и матриц. Создания различных структур в C++. Исследование особенностей использования функций в C++. Исследование перегрузки функций. Область видимости переменных в функциях. Исследование алгоритмов обработки структур в C++. Обработка массивов и матриц. Обработка строк в C++. Составление приложений с инкапсуляцией и полиморфизмом методов классов. Практическое создание скрытых методов и полей, а также полиморфных методов классов. Исследование приложений с различными видами наследования. Исследование различных видов наследования классов.</i>	СРС	30	ПК-4	Л1.1, Л2.1, Л3.2
Курс 5,					
Модуль 2. Разработка кроссплатформенных приложений в среде Qt - 72 (12+51+9) часов					
2.1	Виды наследования. Комбинации доступа. Выбор спецификатора доступа. Уровни наследования. Множественное наследование.	Лек.	2	ПК-4	Л1.1, Л1.2
2.2	Разработка приложений с инкапсуляцией и	ПЗ	2	ПК-4	Л1.1,

	полиморфизмом. <i>Исследование особенностей скрытых методов и полей, а также полиморфных методов классов.</i>				ЛЗ.2
2.3	Полиморфизм и инкапсуляция в приложениях на C++. <i>Модификаторы доступа к объектам класса. Составление полиморфных сигнатур для одноименных методов.</i>	Лек.	4	ПК-4	Л1.1, Л1.2
2.4	Реализация приложений с интерфейсами пользователя в C++. <i>Программная реализация оконных приложений в Qt. Решение практических задач.</i>	ПЗ	2	ПК-4	Л1.1, ЛЗ.1
2.5	Приложения с элементами ООП. <i>Исследование реализации оконных приложений в Qt, содержащих полиморфизм, наследование и инкапсуляцию.</i>	ПЗ	2	ПК-4	Л1.1, ЛЗ.2
2.6	<i>Изучение вариантов составления структур. Простые и управляющие операторы. Изучение команд препроцессора и функций пользователя. Файлы и работа с ними. Библиотеки для работы с комплексными числами. Обработка двоичных файлов. Перегрузка функций и конструкторов. Организация рекурсий в функциях. Знакомство с Qt. Подготовка к работе. Справка и ресурсы. Обзор настроек среды Qt Creator. Создание первого проекта. Структура проекта. Основные типы. Компиляция проекта. Консольный проект Qt. Вывод сообщений. Работа с текстовыми строками в Qt. Классы и файлы в Qt. Контейнерные классы в Qt. Работа с файлами. Создание графического интерфейса средствами Qt. Виджеты (Widgets). Компоновка (Layouts). Создание сигналов (signals) и слотов (slots). Создание элементов графического интерфейса. Класс QObject. События (Events). Обработка событий (Event handling). Создание собственного элемента интерфейса. Программирование формы созданной в Qt Designer. Исследование консольных приложений в Qt. Исследование особенностей создания различных кроссплатформенных приложений. Обработка исключений. Ситуации возникновения исключений. Шаблон обработки исключений. Создание консольных приложений в Qt. Практическая тренировка в создании различных кроссплатформенных приложений.</i>	СРС	31	ПК-4	Л1.1, Л2.1, ЛЗ.2
2.7	Контрольная работа «Разработка кроссплатформенных приложений на C++»	СРС	20	ПК-4	Л1.1, Л2.1, ЛЗ.2, ЛЗ.2
2.8	Экзамен		9	ПК-4	Л1.1-ЛЗ.2, Э1-Э5
Итого – 108 часов					

5. Учебно-методическое и информационное обеспечение дисциплины

5.1 Рекомендуемая литература				
5.1.1. Основная литература				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л1.1	Русанова Я. М., Чердынцева М.И.	C++ как второй язык в обучении приемам и технологиям программирования	Южный федеральный университет, 2010г.	Э1
Л1.2	Гагарина Л. Г., Дорогова Е. Г.	Основы программирования на языке C	ИНФРА-М, 2025г.	Э2
5.1.2 Дополнительная литература				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л2.1	Дорогов В. Г., Дорогова Е.Г.	Основы программирования на языке C	Издательский Дом ФОРУМ, 2023г.	Э3
Л2.2	Кузин А.В., Чумакова Е. В.	Основы программирования на языке Objective-C для iOS	ИНФРА-М, 2025г.	Э4
5.1.3 Учебно-методическое обеспечение для самостоятельной работы обучающихся				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л3.1	П.В. Лобзенко, И.В. Щербань	Разработка кроссплатформенных Приложений C++. Лабораторный практикум	Ростов-на-Дону: Северо-Кавказский филиал МТУСИ, 2018	Э5
Л3.2	П.В. Лобзенко, И.В. Щербань	Разработка кроссплатформенных Приложений C++. Практикум	Ростов-на-Дону: Северо-Кавказский филиал МТУСИ, 2018	Э6
5.2 Электронные образовательные ресурсы				
Э1	https://znanium.ru/catalog/document?id=137430			
Э2	https://znanium.ru/catalog/document?id=449894			
Э3	https://znanium.ru/catalog/document?id=427301			
Э4	https://znanium.ru/catalog/document?id=452754			
Э5	https://skf.mtuci.ru/upload/iblock/e88/106sk4x5ntsvrrss1uowato6z7zf8kuk.pdf			
Э6	https://skf.mtuci.ru/upload/iblock/dba/l1q8hcwydp6cjzmjqk0kwob2w4yea0fg.pdf			
5.3 Программное обеспечение				
СКФ МТУСИ обеспечен необходимым комплектом лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства:				
П.1	MS Visual Studio .NET 2022 (C++), Qt (C++)			
П.2	MS Visio 2010, MS Access 2010, MS Word 2010			

6. Материально-техническое обеспечение дисциплины

6.1 МТО лекционных занятий

1	Лекционная аудитория, оснащенная проектором, ПК (ноутбуками), экраном
6.2 МТО лабораторных работ и практических занятий	
1	Учебные аудитории, оборудованные компьютерной техникой, с возможностью подключения к сети «Интернет», и обеспечением доступа в электронную информационно-образовательную среду СКФ МТУСИ
6.3 МТО рубежных контролей и зачетов	
1	Учебные аудитории, оборудованные компьютерной техникой, с возможностью подключения к сети «Интернет», и обеспечением доступа в электронную информационно-образовательную среду СКФ МТУСИ
6.3 МТО самостоятельной работы обучающихся	
1	Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду СКФ МТУСИ

7. Оценочные материалы	
Оценочные материалы и перечень видов оценочных средств текущего контроля и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.	

8. Методические рекомендации для обучающихся по освоению дисциплины	
Методические рекомендации по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.	

9. Особенности реализации дисциплины (модуля) при обучении инвалидов и лиц с ограниченными возможностями здоровья

Рабочая программа дисциплины (модуля) при необходимости может быть адаптирована для обучения (в том числе с применением дистанционных образовательных технологий) лиц с ограниченными возможностями здоровья, инвалидов. Для этого требуется заявление обучающихся, являющихся лицами с ограниченными возможностями здоровья, инвалидами, или их законных представителей и рекомендации психолого-медико-педагогической комиссии. При обучении лиц с ограниченными возможностями здоровья учитываются их индивидуальные психофизические особенности. Обучение инвалидов осуществляется также в соответствии с индивидуальной программой реабилитации инвалида (при наличии).

Для лиц с нарушением слуха возможно предоставление учебной информации в визуальной форме (краткий конспект лекций; тексты заданий, напечатанные увеличенным шрифтом), на аудиторных занятиях допускается присутствие ассистента, а также сурдопереводчиков и тифлосурдопереводчиков.

Текущий контроль успеваемости осуществляется в письменной форме: обучающийся письменно отвечает на вопросы, письменно выполняет практические задания.

Доклад (реферат) также может быть представлен в письменной форме, при этом требования к содержанию остаются теми же, а требования к качеству изложения материала (понятность, качество речи, взаимодействие с аудиторией и т. д.) заменяются на соответствующие требования, предъявляемые к письменным работам (качество оформления текста и списка литературы, грамотность, наличие иллюстрационных материалов и т.д.).

Промежуточная аттестация для лиц с нарушениями слуха проводится в письменной форме, при этом используются общие критерии оценивания. При необходимости время подготовки к ответу может быть увеличено.

Для лиц с нарушением зрения допускается аудиальное предоставление информации, а также использование на аудиторных занятиях звукозаписывающих устройств (диктофонов и т.д.). Допускается присутствие на занятиях ассистента (помощника), оказывающего обучающимся необходимую техническую помощь.

Текущий контроль успеваемости осуществляется в устной форме.

При проведении промежуточной аттестации для лиц с нарушением зрения тестирование может быть заменено на устное собеседование по вопросам.

Для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата, на аудиторных занятиях, а также при проведении процедур текущего контроля успеваемости и промежуточной аттестации могут быть предоставлены необходимые технические средства (персональный компьютер, ноутбук или другой гаджет); допускается присутствие ассистента (ассистентов), оказывающего обучающимся необходимую техническую помощь (занять рабочее место, передвигаться по аудитории, прочесть задание, оформить ответ, общаться с преподавателем).

10. Дополнения и изменения в Рабочей программе

Лист актуализации рабочей программы дисциплины Разработка кроссплатформенных приложений Б1.В.ДВ.05.01 для использования в 20__/20__ учебном году

Утверждаю

Зам. директора по УР _____
«__» _____ 20__ г.

Направление: **09.03.01. Информатика и вычислительная техника**

Профиль: **Искусственный интеллект и машинное обучение**

Форма обучения: очная, заочная

(Возможны следующие варианты):

а) Рабочая программа действует без изменений.

б) В рабочую программу вносятся следующие изменения:

- 1)
- 2)
- 3)

Разработчик (и): _____
(ФИО, ученая степень, ученое звание)

«__» _____ 20__ г.

Рабочая программа пересмотрена и одобрена на заседании кафедры ИВТ

Протокол № _____ от «__» _____ 20__ г.

Заведующий кафедрой _____

Оценочные материалы для проведения текущего контроля и промежуточной аттестации по дисциплине

1. Оценочные материалы для проведения промежуточной аттестации по дисциплине

1.1 Шкала оценивания компетенций

Шкала оценивания компетенций		
Оценка	Уровень освоения компетенции	Критерии оценивания
«Отлично»	Высокий уровень	Обучающийся показывает всестороннее, систематическое и глубокое знание основного и дополнительного учебного материала, умеет свободно выполнять задания, предусмотренные программой; усвоил основную и знаком с дополнительной рекомендованной литературой; может объяснить взаимосвязь основных понятий дисциплины в их значении для последующей профессиональной деятельности; проявляет творческие способности в понимании, изложении и использовании учебного материала.
«Хорошо»	Повышенный уровень	Обучающийся показывает достаточный уровень знаний в пределах основного учебного материала, без существенных ошибок выполняет предусмотренные в программе задания; усвоил основную литературу, рекомендованную в программе; способен объяснить взаимосвязь основных понятий дисциплины при дополнительных вопросах преподавателя. Допускает не существенные погрешности в ответах, устраняет их без помощи преподавателя.
«Удовлетворительно»	Пороговый уровень	Обучающийся показывает знания основного учебного материала в минимальном объеме, необходимом для дальнейшей учебы; справляется с выполнением заданий, предусмотренных программой, допуская при этом большое количество не принципиальных ошибок; знаком с основной литературой, рекомендованной программой. Допускает существенные погрешности в ответах, но обладает необходимыми знаниями для их устранения под руководством преподавателя.
«Неудовлетворительно»	Минимальный уровень не достигнут	Обучающийся обнаруживает пробелы в знаниях основного учебного материала, допускает принципиальные ошибки в выполнении предусмотренных программой заданий, не знаком с рекомендованной литературой, не может исправить допущенные ошибки. Как правило, оценка «неудовлетворительно» ставится обучающимся, которые не могут продолжить обу-

		чение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей дисциплине.
--	--	--

1.2 Показатели, критерии и шкалы оценивания компетенций

Показатели компетенции	Критерии оценивания	Шкала оценивания
ПК-4 Способен разрабатывать программное обеспечение с применением методов машинного обучения		
Знать:		
методы машинного обучения, приемы их оценивания и использования	Тест 1	Модуль 1 0÷18 «Неудовлетворительно» - 0-5 «Удовлетворительно» - 6-8 «Хорошо» - 9-11 «Отлично» - 12-18
методы машинного обучения, приемы их оценивания и использования	Тест 2	Модуль 2 0÷18 «Неудовлетворительно» - 0-5 «Удовлетворительно» - 6-8 «Хорошо» - 9-11 «Отлично» - 12-18
Уметь:		
- осуществлять оценку и выбор алгоритмов машинного обучения в соответствии с поставленной задачей;	Практическое занятие 1	Модуль 1 0÷12 «Неудовлетворительно» - 0-1 «Удовлетворительно» - 2-5 «Хорошо» - 6-8 «Отлично» - 9-12
- выполнять поиск, подготовку и разметку структурированных и неструктурированных данных для машинного обучения; - разрабатывать программное обеспечение на основе алгоритмов машинного обучения.	Практическое занятие 4	Модуль 2 0÷12 «Неудовлетворительно» - 0-1 «Удовлетворительно» - 2-5 «Хорошо» - 6-8 «Отлично» - 9-12
Владеть:		
технологиями разработки программного обеспечения с применением методов машинного обучения	Практическое занятие 2	Модуль 1 0÷10 «Неудовлетворительно» - 0-2 «Удовлетворительно» - 3-4 «Хорошо» - 5-6 «Отлично» - 7-10
	Практическое занятие 3	Модуль 1 0÷10 «Неудовлетворительно» - 0-2 «Удовлетворительно» - 3-4 «Хорошо» - 5-6 «Отлично» - 7-10
	Практическое занятие 5	Модуль 2 0÷10

		«Неудовлетворительно» - 0-2 «Удовлетворительно» - 3-4 «Хорошо» - 5-6 «Отлично» - 7-10
	Практическое занятие 6	Модуль 2 0÷10 «Неудовлетворительно» - 0-2 «Удовлетворительно» - 3-4 «Хорошо» - 5-6 «Отлично» - 7-10
	Экзамен	0-100 «Неудовлетворительно» - 0-40 «Удовлетворительно» - 41-60 «Хорошо» - 61-80 «Отлично» - 81-100

1.3 Оценочные материалы: типовые контрольные задания, иные материалы

1.3.1 Оценочные материалы текущего контроля для очной формы обучения

Данная дисциплина формирует у обучающегося 1 компетенцию, которая может быть оценена при ответе обучающегося на контрольный вопрос или при выполнении практического задания.

Модуль 1 (50 баллов):

Модуль содержит 6 лекционных и 3 практических занятия. Знание лекционного материала оценивается по результатам выполнения теста, максимальное количество баллов за тест - 18. За выполненные и защищенные практические занятия студент максимально получает 32 балла (12 баллов за первое из них и по 10 баллов – за каждое из оставшихся). Общее максимальное количество баллов за модуль составляет 50 баллов.

Практическое занятие №1. (ПК-4)

Контрольные вопросы к ПЗ.

Контрольные вопросы (ПК-4):

1. Что такое функция (метод) пользователя?
2. Арифметические выражения в C++.
3. Логические выражения в C++.
4. Оператор присваивания в C++.
5. Как оформляются функции в классах?
6. Какие модификаторы доступа существуют при задании функций?
7. Операторы переходов (программирование разветвляющихся процессов) в C++.
8. Операторы организации циклов в C++.
9. Программирование линейной и разветвляющейся структур в C++.

Практическое занятие №2. (ПК-4)

Контрольные вопросы к ПЗ.

1. Как оформление сигнатуры функции влияет на ее использование в программе?
2. Каким образом задается функция не возвращающая (возвращающая) значения?
3. Что такое «тело» функции?
4. Каким образом обращаются к функции при составлении программы?

5. *Что такое модификаторы доступа к функциям (методам)?*
6. *Что такое простое наследование в C++?*
7. *Что такое множественное наследование в C++?*
8. *Каким образом реализуется простое наследование в C++?*
9. *Каким образом реализуется множественное наследование в C++?*

Практическое занятие №3. (ПК-4)

Контрольные вопросы к ПЗ.

1. *Что такое инкапсуляция?*
2. *Для каких объектов используется инкапсуляция и для чего?*
3. *Что такое модификаторы доступа к объектам?*
4. *Как реализуется инкапсуляция в C++?*
5. *Что такое полиморфизм?*
6. *Каким образом реализуется полиморфизм в C++?*

Модуль 2: (50 баллов):

Модуль содержит 6 лекционных и 3 практических занятия. Знание лекционного материала оценивается по результатам выполнения теста, максимальное количество баллов за тест - 18. За выполненные и защищенные практические занятия студент максимально получает 32 балла (12 баллов за первое из них и по 10 баллов – за каждое из оставшихся). Общее максимальное количество баллов за модуль составляет 50 баллов.

Практическое занятие №4. (ПК-4)

Контрольные вопросы к ПЗ.

1. *Поясните назначение основных элементов программы на C++: элемента `#include <iostream>`.*
2. *Поясните назначение основных элементов программы на C++: элемента `using namespace std`;*
3. *Поясните назначение основных элементов программы на C++: элемента `int main () {}`.*
4. *При создании консольного приложения какие команды необходимо выполнить в интегрированной среде Qt?*
5. *Каким образом производится «сборка» проекта в среде Qt?*
6. *Как проект запускается на решение?*
7. *Где и каким образом отображаются результаты выполнения проекта?*

Практическое занятие №5. (ПК-4)

Контрольные вопросы к ПЗ.

1. *Какие библиотеки используются при создании пользовательских интерфейсов?*
2. *Что такое форма и как ее создать?*
3. *Какие инструменты используются при создании форм?*
4. *Где расположены инструменты создания графических интерфейсов?*
5. *Что такое событие? Как создаются события?*
6. *Что такое обработчик события и как он оформляется?*
7. *Как обратиться к методу «не своего» класса из обработчика событий?*
8. *Как выглядит main-метод при создании форм?*
9. *Что такое менеджеры размещения компонентов форм?*
10. *Какие известны основные компоновщики?*

Практическое занятие №6 (ПК-4)

Контрольные вопросы к ПЗ.

1. *Что такое инкапсуляция?*
2. *Для каких объектов используется инкапсуляция и для чего?*
3. *Что такое модификаторы доступа к объектам?*
4. *Как реализуется инкапсуляция в C++?*
5. *Что такое полиморфизм?*
6. *Каким образом реализуется полиморфизм в C++?*

1.3.2. Вопросы и задания, выносимые на экзамен по дисциплине Разработка кроссплатформенных приложений (ПК-4) (промежуточная аттестация):

Вопросы:

1. Кроссплатформенное программирование линейного вычислительного процесса в C++ (ПК-4).
2. Кроссплатформенное программирование разветвляющегося вычислительного процесса в C++(ПК-4).
3. Кроссплатформенное программирование циклического вычислительного процесса в C++(ПК-4).
4. Кроссплатформенный язык программирования C++: назначение, отличия как процедурного языка высокого уровня (ПК-4).
5. Основная структура языка C++. Алфавит. Основные типы данных (ПК-4).
6. Арифметические выражения в C++ (ПК-4).
7. Логические выражения в C++ (ПК-4).
8. Оператор присваивания в C++ (ПК-4).
9. Операторы описания типа в C++ (ПК-4).
10. Операторы ввода информации в C++ (ПК-4).
11. Операторы вывода информации в C++ (ПК-4).
12. Операторы переходов в C++ (ПК-4).
13. Операторы организации циклов в C++ (ПК-4).
14. Кроссплатформенное программирование линейной и разветвляющейся структур в C++ (ПК-4).
15. Кроссплатформенное программирование циклической структуры в C++ (ПК-4).
16. Объектно-ориентированное программирование. Понятия и определения. Инкапсуляция, полиморфизм и наследование (ПК-4).
17. Кроссплатформенное программирование. Основные принципы структурного подхода. Метод пошаговой детализации (ПК-4).
18. Какие этапы включает компьютерная технология решения задачи? Дайте их краткую характеристику (ПК-4).
19. Что такое структура (ПК-4).
20. Сформулируйте основную задачу программирования (ПК-4).
21. Что такое итерация и рекурсия в кроссплатформенном программировании. Поясните эти понятия на примере (ПК-4).

22. Алгоритм создания кроссплатформенных приложений с интерфейсом пользователя в C++(ПК-4).

Задания:

1. Числа S, T ввести с клавиатуры в функции main и вычислить выражение: $G(12, S) + G(T, S) - G(2S-1, S*T)$, где в класс Klass1 входит функция - $G(A, B) = (2A+B^3)/(2AB+5B)F(A,B)$, а в класс Klass2 входит функция - $F(A,B) = A \sin(B/A)$ (ПК-4).
2. Действительные числа A,B вводятся с клавиатуры в функции vvod1, а действительные числа C,D- в функции vvod2. Эти функции входят в класс Q1. Составить функцию Pr, в которой можно было бы выяснить, можно ли уместить прямоугольник со сторонами A,B внутри прямоугольника со сторонами C,D. Функцию Pr включить в класс Q2 (ПК-4).
3. Составить функцию определения среднего арифметического положительных чисел, введенных с клавиатуры. Эту функцию включить в класс CL2. Всего ввести N различных чисел в функции, входящей в CL1. Напечатать адрес ячейки, в которой хранится значение последнего вводимого числа (ПК-4).
4. Составить класс вывода на печать положительных значений выражения $y = \sin(x^5) + 5\cos^3(x-2)$ и адресов ячеек, в которых они хранятся, для x изменяющегося на отрезке $[-5, 12]$ с шагом 1,2. Использовать цикл с предусловием. X ввести в функции класса M1, а выражение вычислять в функции класса B2 (ПК-4).
5. Записать класс для вычисления функции $F(X,Y) = (2X^3 - 4*X^2 + X + 1)/(9*Y^3 + Y + 4) + 3*Y^2 + 5*Y$. Вычислить при $X=5.1, Y=6.2$ вводимых клавиатуры в функции main (ПК-4).
6. Определить, лежит ли точка D(c,b), где c= внутри прямоугольника со сторонами 5 и 10 (a_1, a_2, a_3 - произвольные числа)? Оформить в виде класса с конструктором, через который ввести числа a_1, a_2, a_3 (ПК-4).
7. Для арифметической прогрессии 4, 9, 14, 19... найти первые n членов этой прогрессии. Использовать цикл с параметром - FOR. Оформить в виде класса (ПК-4).
8. Вывести на печать значения функции $z = \ln(x) + \operatorname{tg}(2x)$, большие 1, для x изменяющегося на отрезке $[3, 8]$ с шагом 0,9. Использовать цикл с предусловием. Оформить в виде функции пользователя (ПК-4).
9. Дано натуральное число N. Получить наименьшее число вида $2k$, большее N. Использовать цикл с постусловием. Оформить в виде класса (ПК-4).
10. Даны числа $x = 1, 2, \dots, N$. Вычислить: $3 * P(X+3) * P(X)$ для x, введенных с клавиатуры. Использовать функцию: $P(X) = 10 * X^3 - 14 * X^2 + 12 * X - 2$ и включить ее в класс (ПК-4).
11. Даны круг радиуса R и квадрат со стороной A. Определить можно ли вписать их друг в друга. Оформить в виде функции пользователя (ПК-4).
12. Даны числа S, T. Составить класс для вычисления выражения $F(T, -2S, 1.17) + F(2.2, T, S-T)$, в который входит функция: $F(A, B, C) = (2A - B - \sin(C)) / (5 + C)$ (ПК-4).
13. Найти n членов последовательности $x_1 = x_2 = x_3 = 1; x_k = x_{k-1} + x_{k-3}$. Использовать цикл с параметром. Оформить в виде класса (ПК-4).

14. Вывести на печать значения функции $z = \sin(x) + \cos(x)$, находящиеся в интервале $(-0,2; 0,8)$ для x изменяющегося на отрезке $[4,-6]$ с шагом 0,91. Использовать цикл с предусловием. Оформить в виде функции пользователя (ПК-4).
15. Найти среднее арифметическое отрицательных чисел введенных с клавиатуры. Всего ввести N различных чисел. Оформить в виде класса (ПК-4).
16. Составить программу, состоящую из 2-х классов: базового(предка) и производного (наследника) и передать значение переменной из базового класса в производный класс (ПК-4).
17. Задав функцию пользователя, вывести на печать среднее арифметическое элементов двух одномерных массивов, введенных с клавиатуры. Массив передается функции в качестве параметра (ПК-4).
18. Для введенных с клавиатуры чисел найти сумму положительных из них. При вводе отрицательного числа суммирование завершить. Оформить в виде класса (ПК-4).
19. Даны числа A, B, C . Получить с использованием функции пользователя наименьшее значение. Числа передаются функции из основной программы в качестве параметров (ПК-4).
20. Напечатать таблицу значений функции $Y = \lg(x/b) + x/(b-2)$ для x , изменяющегося от 0 до 10 с шагом 1 (b - произвольное число). Использовать цикл с параметром. Оформить в виде класса (ПК-4).
21. Найти n членов последовательности $x_1 = x_2 = x_3 = 1; x_k = x_{k-1} + x_{k-3}$. Использовать цикл с параметром. Оформить в виде функции пользователя (ПК-4).

1.4 Оценочные материалы для заочной формы обучения

Практическое занятие №1 (ПК-4)

Контрольные вопросы к ПЗ.

1. Как оформление сигнатуры функции влияет на ее использование в программе?
2. Каким образом задается функция не возвращающая (возвращающая) значения?
3. Что такое «тело» функции?
4. Каким образом обращаются к функции при составлении программы?
5. Что такое модификаторы доступа к функциям (методам)?
6. Что такое простое наследование в C++?
7. Что такое множественное наследование в C++?
8. Каким образом реализуется простое наследование в C++?
9. Каким образом реализуется множественное наследование в C++?

Практическое занятие №2 (ПК-4)

Контрольные вопросы к ПЗ.

1. Что такое инкапсуляция?
2. Для каких объектов используется инкапсуляция и для чего?
3. Что такое модификаторы доступа к объектам?
4. Как реализуется инкапсуляция в C++?
5. Что такое полиморфизм?
6. Каким образом реализуется полиморфизм в C++?

Практическое занятие №3 (ПК-4)

Контрольные вопросы к ПЗ.

1. Какие библиотеки используются при создании пользовательских интерфейсов?

2. Что такое форма и как ее создать?
3. Какие инструменты используются при создании форм?
4. Где расположены инструменты создания графических интерфейсов?
5. Что такое событие? Как создаются события?
6. Что такое обработчик события и как он оформляется?
7. Как обратиться к методу «не своего» класса из обработчика событий?
8. Как выглядит main-метод при создании форм?
9. Что такое менеджеры размещения компонентов форм?
10. Какие известны основные компоновщики?

Практическое занятие №4 (ПК-4)

Контрольные вопросы к ПЗ.

1. Что такое инкапсуляция?
2. Для каких объектов используется инкапсуляция и для чего?
3. Что такое модификаторы доступа к объектам?
4. Как реализуется инкапсуляция в C++?
5. Что такое полиморфизм?
6. Каким образом реализуется полиморфизм в C++?

Тема контрольной работы:

Разработка кроссплатформенного приложения на C++

Целью контрольной работы является закрепление теоретического материала дисциплины, а также освоение особенностей разработки кроссплатформенных приложений на C++.

Контрольная работа состоит из 3-х заданий, составляющих исходный консольный код на языке C++. Задания необходимо разместить в классах проекта и составить из них кроссплатформенное приложение с определенным механизмом наследования.

Экзамен по дисциплине у студентов заочной формы обучения проводится письменно по классической методике. Вопросы сгруппированы в билетах. В каждом билете содержится по 1 теоретическому и по одному практическому вопросу для оценки компетенции. Количество билетов должно быть не менее числа студентов в группе. Студенты готовят письменные ответы на вопросы. Перед ответом на вопросы экзамена проводится собеседование по выполненной контрольной работе.

Критерии оценки контрольной работы

№п/п	Оцениваемые параметры	Оценка в баллах
1	Выполнение задания контрольной работы:	
	- Задание выполнено в полном объеме, каждый рассматриваемый вопрос раскрыт полностью;	5
	- Задание выполнено в полном объеме, один или несколько рассматриваемых вопросов раскрыты неполно;	3
	- Один или несколько пунктов задания не выполнены или выполнены не в полном объеме	0
2	Оформление результатов работы:	
	- контрольная работа оформлена в полном соответствии с ЕСКД;	5
	- контрольная работа оформлена с незначительными отклонениями от ЕСКД;	3
	- оформление контрольной работы не соответствует требованиям ЕСКД.	0

№п/п	Оцениваемые параметры	Оценка в баллах
3	Качество ответов на вопросы: - отвечает на вопросы; - не может ответить на большинство вопросов; - не может четко ответить на вопросы.	5 3 0

Обучающийся, набравший менее 3 баллов хотя бы по одному из пунктов 1 – 3, получает по контрольной работе «незачет». Обучающийся, набравший от 9 баллов и выше – «зачет».

Экзаменационные вопросы такие же, как и для студентов очной формы обучения

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

2.1. Порядок и методика проведения текущего контроля и промежуточной аттестации.

Предусмотрены следующие виды контроля:

- текущий контроль по каждому модулю – в форме выполнения тестового контроля для оценки теоретических знаний;
- текущий контроль по каждому модулю – в форме отчетов по практическим занятиям для оценки практических навыков;
- промежуточная аттестация по дисциплине – в форме экзамена.

Текущий контроль успеваемости в форме защиты лабораторных и практических занятий проводится в два этапа:

- 1-й этап: допуск к выполнению практического занятия – проводится в форме письменной «летучки» (5-10 мин) с целью контроля готовности к решению практических заданий;
- 2-й этап выполняется по окончании каждого практического занятия в форме индивидуального собеседования по выполненным расчетам. Проводится с целью контроля закрепления теоретической части материала и степени отработки обучающимся практических навыков в решении практических задач.

С целью повышения качества обучения за счет побуждения обучающихся к активной текущей учебной работе, четкого и оперативного контроля всего хода учебного процесса, снижения роли случайных и субъективных факторов при оценивании учебной деятельности обучающихся в образовательном процессе реализована модульно-рейтинговая система.

Правила ее использования описаны в «Положении об МРС».

Набранные обучающимися баллы могут быть переведены в оценку:

- «неудовлетворительно» - от 0% до 40% от максимального количества баллов;
- «удовлетворительно» - от 41% до 60% максимального количества баллов;
- «хорошо» - от 61% до 80% максимального количества баллов;
- «отлично» - от 81% до 100% максимального количества баллов.

Для получения экзамена обучающемуся достаточно набрать 41 и более баллов.

Соотношения максимального количества баллов, полученных обучающимся по блокам модулей, показаны в Таблице.

Таблица - Распределение баллов по блокам модулей дисциплины «Разработка кроссплатформенных приложений»

Модуль	Всего баллов (Максимальное значение)	Теоретический блок	Практический блок (Распределение баллов по занятиям)
Модуль 1	50	18	32
Модуль 2	50	18	32
Экзамен		100	

На экзамене производится оценка тех компетенций, которые должны быть в той или иной форме освоены в процессе изучения. Рекомендуется формировать вопросы в билетах таким образом, чтобы преподаватель смог оценить все компетенции данной дисциплины.

2.2 Методика проведения экзамена в группах очной и заочной формы обучения

Экзамен по дисциплине «Разработка кроссплатформенных приложений» у обучающихся очной и заочной форм обучения проводится письменно по классической методике. Вопросы сгруппированы в билетах. В каждом билете содержится по 2 вопроса, скомпонованные таким образом, чтобы преподаватель смог оценить все компетенции данной дисциплины. Количество билетов должно быть не менее числа обучающихся в группе. Обучающиеся готовят письменные ответы на вопросы.

3. Промежуточная аттестация

Промежуточная аттестация проводится в форме экзамена. Экзамен проводится по расписанию в устном виде по экзаменационным билетам. Результаты аттестации заносятся в ведомость и зачетную книжку обучающегося.

Образец экзаменационного билета

	МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ Северо-Кавказский филиал ордена Трудового Красного Знамени федерального государственного бюджетного образовательного учреждения высшего образования «Московский технический университет связи и информатики»	Утверждаю Зав. кафедрой «ИВТ» _____ Соколов С.В. «___» _____ 20__ г.
Направление подготовки: 09.03.01 Информатика и вычислительная техника Курсы: 3, 4 Дисциплина: Разработка кроссплатформенных приложений C++		
Билет №1 1. Комбинации доступа к объектам классов в языках программирования. Выбор спецификатора доступа. 2. Разработать приложение: Числа S, T ввести с клавиатуры в конструкторе класса и вычислить выражение: $G(12, S) + G(T, S) - G(2S - 1, S * T)$, где в класс Klass1 входит функция - $G(A, B) = (2A + B^3) / (2AB + 5B)F(A, B)$, а в класс Klass2 входит функция - $F(A, B) = \text{Asin}(B/A)$. Результат вывести в файл. «___» _____ 20__ г. Доцент каф. «ИВТ» _____ Лобзенко П.В		

Один комплект отпечатанных экзаменационных билетов, подписанных преподавателем кафедры и утвержденных заведующим кафедрой хранится у заведующего кафедрой, другой комплект – у преподавателя, ведущего дисциплину.

4 Тестовые задания для проведения оценки сформированности компетенций (ПК-4)

Один комплект тестовых вопросов с указанием правильных ответов хранится у заведующего кафедрой, другой комплект – у преподавателя, ведущего дисциплину.

Тестовые задания позволяют оценить уровень сформированности компетенции ПК-4.

Задания распределены по блокам в соответствии с уровнем сложности. Каждый блок содержит номер задания и текст задания.

Базовый уровень содержит примерно 50 % заданий.

Он формируется из заданий с выбором одного или нескольких верных ответов из предложенных или воспроизведения фактического материала (терминология, факты, классификация, параметры и др.).

Повышенный уровень в среднем составляет 35 % заданий.

В нем используются задания на сопоставление, сравнение, установление последовательности;

Высокий уровень в среднем составляют – 15 % заданий.

В задании используются: решения нетиповых задач, алгоритмы, доказательства, задания с развернутым ответом – определения, перечисление, изображение схемы, структуры и др.

Количество заданий равно 20.

ПК-4: Способен разрабатывать программное обеспечение с применением методов машинного обучения

БЛОК А (базовый уровень) – Задание с выбором одного или нескольких верных ответов из предложенных

Инструкция по тестам Блока А: Прочитайте текст и выберите один или несколько правильных ответов

№ задания	Тексты заданий
1	Программа, которая переводит исходную программу на языке высокого уровня в эквивалентную ей программу в кодах машины, называется: 1) транслятор 2) компилятор 3) интерпретатор 4) сканер
2	Какой или какие из указанных языков объектно-ориентированные? 1) Eiffel 2) C++ 3) Java 4) Все варианты ответов
3	Какой фрагмент есть общей записью оператора множественного выбора – switch? 1) <pre>switch (switch_expression) { case constant1: statement1; [break;] case constant2: statement2; [break;] case constantN: statementN; [break;] [default: statement N+1;]}</pre> 2) <pre>switch (switch_expression) { case constant1: statement1; [break;] case constant2: statement2; [break;] case constantN: statementN; [break;]}</pre>

№ задания	Тексты заданий
	<pre>[else: statement N+1;] }</pre> 3)switch (switch_expression) <pre>{ case constant1, case constant2: statement1; [break;] case constantN: statementN; [break;] [default: statement N+1;] }</pre>
4	Какая из форм является правильной записью цикла do while? 1) // форма записи оператора цикла do while: <pre>do // начало цикла do while { /*блок операторов*/; } while { /*условие выполнения цикла*/ } // конец цикла do while</pre> 2) // форма записи оператора цикла do while: <pre>do // начало цикла do while (/*блок операторов*/;) while (/*условие выполнения цикла*/); // конец цикла do while</pre> 3) // форма записи оператора цикла do while: <pre>do // начало цикла do while { /*блок операторов*/; } while (/*условие выполнения цикла*/) // конец цикла do while</pre>
5	Какие среды программирования (IDE) предназначены для разработки кроссплатформенных программных средств? 1) Unity, QT Creator, Visual Studio 2) MVS, Code::Blocks, QT Creator, RAD Studio, MathCAD 3) MVS, Code::Blocks, QT Creator, AutoCAD, Eclipse
6	Чему будет равна переменная a, после выполнения этого кода <pre>int a; for(a = 0; a < 10; a++) {}?</pre> 1) 1 2) 9 3) 10

№ задания	Тексты заданий
7	Какая строчка подключает файл <code>iostream</code> в программу на C++ : 1) <code>include #iostream,h;</code> 2) <code>include (iostreamh)</code> 3) <code>#include < iostream></code>
8	Каков результат работы следующего фрагмента кода? <pre>int x = 0; switch(x) { case 1: cout << "Один"; case 0: cout << "Ноль"; case 2: cout << "Привет мир"; }</pre> 1) НольПривет мир 2) Привет мир 3) Один 4) Ноль
9	Цикл с предусловием? 1) <code>do while</code> 2) <code>for</code> 3) <code>while</code>
10	Какие служебные символы используются для обозначения начала и конца блока кода в C++? 1. <code>{ }</code> 2. <code>< ></code> 3. <code>()</code> 4. <code>begin end</code>

БЛОК Б (повышенный уровень) – Задание закрытого типа на установление соответствия

Инструкция: Прочитайте текст и установите соответствие или последовательность

№ задания	Тексты заданий
11	Различают следующие варианты разработки приложений (запишите верные соответствия

№ задания	Тексты заданий								
	вида: 1-1, 2-2 и т.д.): <table> <tr> <td>1. кроссплатформенные конвертированные</td><td>1. код на языке C++ создается в кроссплатформенном редакторе и выгружается под нужную платформу</td></tr> <tr> <td>2. кроссплатформенные разработанные</td><td>2. код на языке C++ интерпретируется посредством оболочки-интерпретатора</td></tr> <tr> <td>3. нативные</td><td>3. код на языке C++ содержит ветвления, операторы повторов и подпрограммы</td></tr> <tr> <td>4. структурные</td><td>4. код на языке C++ пишется в редакторе для определенной платформы</td></tr> </table>	1. кроссплатформенные конвертированные	1. код на языке C++ создается в кроссплатформенном редакторе и выгружается под нужную платформу	2. кроссплатформенные разработанные	2. код на языке C++ интерпретируется посредством оболочки-интерпретатора	3. нативные	3. код на языке C++ содержит ветвления, операторы повторов и подпрограммы	4. структурные	4. код на языке C++ пишется в редакторе для определенной платформы
1. кроссплатформенные конвертированные	1. код на языке C++ создается в кроссплатформенном редакторе и выгружается под нужную платформу								
2. кроссплатформенные разработанные	2. код на языке C++ интерпретируется посредством оболочки-интерпретатора								
3. нативные	3. код на языке C++ содержит ветвления, операторы повторов и подпрограммы								
4. структурные	4. код на языке C++ пишется в редакторе для определенной платформы								
12	Найдите соответствие сред разработки и приложений на языке C++: <table> <tr> <td>1) Среда разработки Qt</td><td>А) универсальные приложения для разных платформ</td></tr> <tr> <td>2) Среда разработки Visual Studio</td><td>Б) приложения под одну платформу</td></tr> </table>	1) Среда разработки Qt	А) универсальные приложения для разных платформ	2) Среда разработки Visual Studio	Б) приложения под одну платформу				
1) Среда разработки Qt	А) универсальные приложения для разных платформ								
2) Среда разработки Visual Studio	Б) приложения под одну платформу								
13	Различают разновидности полиморфизма в программах на языке C++: <table> <tr> <td>1. перегрузка названий методов</td><td>1. имеется исходный виртуальный метод, состав которого можно изменять в последующих реализациях</td></tr> <tr> <td>2. эт-хок через перезаписываемые методы</td><td>2. полностью схожие сигнатуры методов различаются прямым указанием класса, где они содержатся</td></tr> <tr> <td>3. второго рода</td><td>3. сигнатуры методов различаются по составу, типу и порядку следования параметров</td></tr> </table>	1. перегрузка названий методов	1. имеется исходный виртуальный метод, состав которого можно изменять в последующих реализациях	2. эт-хок через перезаписываемые методы	2. полностью схожие сигнатуры методов различаются прямым указанием класса, где они содержатся	3. второго рода	3. сигнатуры методов различаются по составу, типу и порядку следования параметров		
1. перегрузка названий методов	1. имеется исходный виртуальный метод, состав которого можно изменять в последующих реализациях								
2. эт-хок через перезаписываемые методы	2. полностью схожие сигнатуры методов различаются прямым указанием класса, где они содержатся								
3. второго рода	3. сигнатуры методов различаются по составу, типу и порядку следования параметров								
14	Найдите соответствие модификаторов доступа их назначению при инкапсуляции в приложениях на языке C++: <table> <tr> <td>1) privet</td><td>А) доступно в любом классе</td></tr> <tr> <td>2) public</td><td>Б) доступно только в своем классе</td></tr> <tr> <td>3) protected</td><td>С) доступно в классе 1-го наследника</td></tr> </table>	1) privet	А) доступно в любом классе	2) public	Б) доступно только в своем классе	3) protected	С) доступно в классе 1-го наследника		
1) privet	А) доступно в любом классе								
2) public	Б) доступно только в своем классе								
3) protected	С) доступно в классе 1-го наследника								
15	Найдите соответствие типов наследования и их определений: <table> <tr> <td>1) множественное</td><td>А) класс-потомок наследуется от более чем одного класса-предка</td></tr> <tr> <td>2) единичное</td><td>Б) класс-потомок наследуется от одного класса-предка</td></tr> </table>	1) множественное	А) класс-потомок наследуется от более чем одного класса-предка	2) единичное	Б) класс-потомок наследуется от одного класса-предка				
1) множественное	А) класс-потомок наследуется от более чем одного класса-предка								
2) единичное	Б) класс-потомок наследуется от одного класса-предка								

№ задания	Тексты заданий				
16	Найдите соответствие определений: <table border="1"> <tr> <td>1. Интерпретатор</td><td>А. Программа-кодировщик, переводящая код программы в машинные коды «заранее»</td></tr> <tr> <td>2. Компилятор</td><td>Б. Программа-кодировщик, переводящая код программы в машинные коды «на лету»</td></tr> </table>	1. Интерпретатор	А. Программа-кодировщик, переводящая код программы в машинные коды «заранее»	2. Компилятор	Б. Программа-кодировщик, переводящая код программы в машинные коды «на лету»
1. Интерпретатор	А. Программа-кодировщик, переводящая код программы в машинные коды «заранее»				
2. Компилятор	Б. Программа-кодировщик, переводящая код программы в машинные коды «на лету»				
17	В кодах на C++ возможно выделение памяти (найдите соответствие): <table border="1"> <tr> <td>1. Динамическое</td><td>А. после компиляции</td></tr> <tr> <td>2. Статическое</td><td>Б. до компиляции</td></tr> </table>	1. Динамическое	А. после компиляции	2. Статическое	Б. до компиляции
1. Динамическое	А. после компиляции				
2. Статическое	Б. до компиляции				

БЛОК В (высокий уровень) – Задание открытого типа с развернутым ответом

Инструкция: Прочитайте текст и заполните пропуски

№ задания	Тексты заданий
18	Кроссплатформенные разработки – это коды, используемые на более чем . _____ одной платформе
19	Программные оболочки (или «обертки») служат для . _____ . исходного кода программы в кроссплатформенный
20	Для кроссплатформенной конвертации кода на языке C++ в редакторе Visual Studio используется технология CLR – это Common. _____ . Runtime

Система оценивания выполнения тестовых заданий по компетенции ОПК-1

Номер задания	Указания по оцениванию	Результат оценивания (баллы, полученные за выполнение задания/характеристика правильности ответа)
Задания 1-10	Задания с выбором одного или нескольких верных ответов из предложенных считается верным, если правильно указана цифра или цифры, означающие верные ответы.	Совпадение с верным ответом оценивается 1 баллом; неверный ответ или его отсутствие - 0 баллов.
Задания 11-17	Задания закрытого типа на установление соответствия считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)	Полное совпадение с верным ответом оценивается 1 баллом; неверный ответ или его отсутствие - 0 баллов.

Задания 18-20	Задание открытого типа с развернутым ответом считается верным, если ответ совпадает с эталонным по содержанию и полноте.	Полный правильный ответ на задание оценивается 3 баллами; если допущена одна ошибка/неточность/ответ правильный, но не полный - 1 балл, если допущено более одной ошибки/ответ неправильный/ответ отсутствует- 0 баллов.
------------------	--	--

Методика проведения тестирования

Максимальное время выполнения теста - 60 минут.

Количество тестовых заданий в тесте не менее 20.

Для проведения контроля остаточных знаний компьютер формирует в случайном порядке 20 тестовых заданий.

При прохождении тестового контроля знаний студент должен выбрать один или несколько правильных ответов в каждом тестовом задании.

В случае правильного ответа на 16 и более тестовых заданий (81-100% правильных ответов) студент подтверждает освоение компетенций с уровнем «отлично». При ответе на 12-15 тестовых заданий (61-80% правильных ответов) студент подтверждает освоение компетенций с уровнем «хорошо». При ответе на 8-14 тестовых заданий (41-60% правильных ответов) студент подтверждает освоение компетенций с уровнем «удовлетворительно». При ответе менее чем на 8 тестовых заданий (0-40% правильных ответов) студент не подтверждает необходимый уровень знаний и оценивается «неудовлетворительно».

Методические рекомендации по освоению дисциплины

1. Методические рекомендации преподавателю

Дисциплина «Разработка кроссплатформенных приложений» включает в себя:

- лекционные занятия – 24 часа;
- практические занятия – 24 часа;
- контрольную работу;
- промежуточную аттестацию – экзамен.

Перед началом изучения дисциплины преподаватель должен ознакомить обучающихся с рабочей программой и оценочными материалами по дисциплине, с видами учебной и самостоятельной работы, перечнем литературы и интернет-ресурсов, с формами текущей и промежуточной аттестации, с критериями оценки качества знаний для итоговой оценки по дисциплине.

При проведении лекций преподаватель:

- 1) формулирует тему и цель занятия, объявляет учебные вопросы;
- 2) излагает основные теоретические положения;
- 3) с помощью технических средств обучения и/или под запись дает определения основных понятий, расчетных формул;
- 4) проводит примеры из отечественного и зарубежного опыта, дает текущие статистические данные для наглядного и образного представления изучаемого материала;
- 5) в конце занятия выдает вопросы для самостоятельного изучения.

Во время выполнения заданий в учебной аудитории обучающийся может консультироваться с преподавателем, определять наиболее эффективные методы решения поставленных задач. Если какая-то часть задания остается не выполненной, обучающийся может продолжить её выполнение во время внеаудиторной самостоятельной работы.

Для оценки полученных знаний и освоения учебного материала по каждому разделу и в целом по дисциплине преподаватель использует формы текущего, промежуточного и итогового контроля знаний обучающихся.

2. Методические рекомендации обучающимся

Приступая к изучению новой учебной дисциплины, обучающиеся должны ознакомиться с рабочей программой, оценочными материалами, учебной, научной и методической литературой, имеющейся в ЭИОС СКФ МТУСИ.

Глубина усвоения дисциплины зависит от активной и систематической работы обучающегося на лекциях и практических занятиях, а также в ходе самостоятельной работы, по изучению рекомендованной литературы.

2.1 Методические указания для обучающихся по лекционным занятиям

Важно сосредоточить внимание на содержании лекции. Это поможет лучше воспринимать учебный материал и уяснить взаимосвязь проблем по всей дисциплине.

Основное содержание лекции целесообразнее записывать в тетради в виде ключевых фраз, понятий, тезисов, обобщений, схем, опорных выводов. Необходимо обращать внимание на термины, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставлять в конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющей материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

С целью уяснения теоретических положений, разрешения спорных ситуаций необходимо задавать преподавателю уточняющие вопросы. Для закрепления содержания лекции в памяти, необходимо

во время самостоятельной работы внимательно прочесть свой конспект и дополнить его записями из учебников и рекомендованной литературы.

Конспектирование читаемых лекций и их последующая доработка способствует более глубокому усвоению знаний, и поэтому являются важной формой учебной деятельности обучающихся.

2.2 Методические указания для обучающихся по подготовке к практическим занятиям

Целью практических занятий является закрепление теоретических знаний, полученных при изучении дисциплины.

При подготовке к практическому занятию целесообразно выполнить следующие рекомендации:

- изучить основную литературу; ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях: журналах, газетах и т.д.;
- при необходимости доработать конспект лекций. При этом учесть рекомендации преподавателя и требования учебной программы.

При выполнении практических занятий основным методом обучения является самостоятельная работа обучающегося под управлением преподавателя. На них пополняются теоретические знания обучающихся, их умение творчески мыслить, анализировать, обобщать изученный материал, проверяется отношение обучающихся к будущей профессиональной деятельности.

Оценка выполненной работы осуществляется преподавателем комплексно: по результатам выполнения заданий, устному сообщению и оформлению работы.

После подведения итогов занятия обучающийся обязан устранить недостатки, отмеченные преподавателем при оценке его работы.

2.3 Методические указания для обучающихся по подготовке к контрольной работе

Контрольная работа для заочной формы обучения выполняется в виде завершённого кроссплатформенного проекта на C++. Таким образом, полученные теоретические знания реализуются на практике каждым обучающимся.

Для подготовки и выполнения контрольной работы используется методическая разработка: П.В. Лобзенко, Щербань И.В., Разработка кроссплатформенных приложений C++, практикум для студентов заочной форм обучения направление подготовки –09.03.01 «Информатика и вычислительная техника», размещенная на сайте СКФ МТУСИ в разделе «Методические материалы».
<https://skf.mtuci.ru/upload/iblock/dba/l1q8hcwydp6cjmjqk0kwob2w4yea0fg.pdf>

3. Методические рекомендации для обучающихся по самостоятельной работе

Самостоятельная работа обучающихся является составной частью учебной работы и имеет целью закрепление и углубление полученных знаний и навыков, поиск и приобретение новых знаний, в том числе с использованием автоматизированных обучающих курсов (систем), а также выполнение учебных заданий, подготовку к предстоящим занятиям, зачетам и экзаменам.

Самостоятельная работа во внеаудиторное время включает в себя:

- повторение лекционного материала;
- подготовку к практическим занятиям;
- изучение учебной и научной литературы;
- решение задач, выданных на практических занятиях;
- подготовку к тестированию;
- выделение наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями кафедры на их еженедельных консультациях.

- проведение самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач, тестов по отдельным вопросам изучаемой дисциплины.

Перед выполнением внеаудиторной самостоятельной работы преподаватель проводит инструктаж (консультацию) с определением цели задания, его содержания, сроков выполнения, основных требований к результатам работы, критериев оценки, форм контроля и перечня источников и литературы.

Обычно постановку задачи обучаемым на проведение самостоятельной работы преподаватель осуществляет на одном из занятий, предшествующем данному.

Методику самостоятельной работы все обучаемые выбирают индивидуально.