Автономная некоммерческая организация высшего образования
«МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ УНИВЕРСИТЕТ»

Рабочая программа дисциплины

Объектно-ориентированное программирование

Направление подготовки	Информационные системы и технологии
Код	09.03.02
Направленность (профиль)	Проектирование, разработка и сопровождение информационных систем
Квалификация выпускника	бакалавр

Москва
2023

1. Перечень кодов компетенций, формируемых дисциплиной в процессе освоения образовательной программы

Группа компетенций	Категория компетенций	Код
Профессиональные	-	ПК-5

2. Компетенции и индикаторы их достижения

Код компетенции	Формулировка компетенции	Индикаторы достижения компетенции
ПК-5	Способен выполнять работы по созданию (модификации) и сопровождению ИС.	**ПК-5.1.** Типовое проектирование информационных систем, а также различных моделей информационных систем и проектных спецификаций; Программные прототипы решения прикладных задач. **ПК-5.2.** Разработка ИС с учетом требований заказчика, на основе стандартов к проектированию информационных систем. Модификация существующих ИС для улучшения их функциональности и производительности. **ПК-5.3.** Способность разрабатывать мобильные приложения и работать с Интернет вещами **ПК-5.4.** Знать и уметь работать с технологиями искусственного интеллекта и инструментальными средствами разработки интеллектуальных программных систем. **ПК-5.5.** Верификация структуры программного кода ИС относительно архитектуры ИС. **ПК-5.6.** Создание пользовательские интерфейсы с учетом UX/UI принципов для повышения удобства использования ИС. **ПК-5.7.** Осуществляет поиск, анализ, программную реализацию математических моделей и алгоритмов интеллектуальной обработки данных.

3. Описание планируемых результатов обучения по дисциплине

3.1. Описание планируемых результатов обучения по дисциплине

Планируемые результаты обучения по дисциплине представлены дескрипторами (знания, умения, навыки).

Дескрипторы по дисциплине	Знать	Уметь	Владеть
Код компетенции	**ПК-5**		

	-современные методы разработки и реализации алгоритмов на базе языков программирования и пакетов прикладных программ; - основы современных систем управления базами данных, теории баз данных; - формальные методы, технологии и инструменты разработки программного обеспечения и баз данных; - основы программирования, современные объектно-ориентированные языки программирования; современные структурные языки программирования, языки современных бизнес-приложений; - современные методики тестирования разрабатываемых ИС: инструменты и методы модульного тестирования, инструменты и методы тестирования нефункциональных и функциональных характеристик ИС.	-разрабатывать алгоритмы и программы на базе языков программирования и пакетов прикладных программ, пригодные для практического применения; - кодировать на языках программирования; - тестировать результаты кодирования;	- приемами разработки алгоритмов и программ на базе языков программирования и пакетов прикладных программ, пригодных для практического применения.

4. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Объектно-ориентированное программирование» относится к части, формируемой участниками образовательных отношений учебного плана ОПОП.

Данная дисциплина взаимосвязана с другими дисциплинами, такими как «Методы и компьютерные технологии имитационного моделирования», «Разработка программных приложений», «Программная инженерия» и др.

В рамках освоения программы бакалавриата выпускники готовятся к решению задач

профессиональной деятельности следующих типов: научно-исследовательский, производственно-технологический, организационно-управленческий, проектный.

Профиль (направленность) программы установлена путем ее ориентации на сферу профессиональной деятельности выпускников: проектирование, разработка и сопровождение информационных систем.

5. Объем дисциплины

Виды учебной работы	***Формы обучения***
	Очная
Общая трудоемкость: зачетные единицы/часы	10/360
Контактная работа:	
Занятия лекционного типа	68
Занятия семинарского типа	84
Промежуточная аттестация: **зачет с оценкой, экзамен.**	18,15
Самостоятельная работа (СРС) **В том числе курсовая работа**	189,85 10

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

6.1. Распределение часов по разделам/темам и видам работы

6.1.1. Очная форма обучения

№ п/п	**Раздел/тема**	**Виды учебной работы (в часах)**						
		Контактная работа						**Самостоятельная работа**
		Занятия лекционного типа		**Занятия семинарского типа**				
		Лекции	*Иные учебные занятия*	*Практические занятия*	*Семинары*	*Лабораторные работы*	*Иные*	
1.	Объекты ООП.	4		4				9,85
2.	Пространства имен.	4		4				8
3.	Модификаторы.	4		4				8
4.	Наследование.	4		4				8
5.	Абстрактные классы.	4		4				8
6.	Обработка исключений.	4		4				8
7.	Делегаты, события и лямбды.	4		4				10
8.	Интерфейсы.	4		4				10
9.	Дополнительные возможности ООП в C#.	4		4				10
10.	Коллекции	4		4				10
11.	Работа со строками	4		4				10
12.	Дополнительные классы	4		4				10

	и структуры .NET							
13.	Многопоточность	4		4				10
14.	Асинхронное программирование	4		4				10
15.	LINQ	4		4				10
16.	Рефлексия. Dynamic Language Runtime. Сборка мусора, управление памятью и указатели.	2		6				10
17.	Работа с потоками и файловой системой	2		6				10
18.	Сериализация.	2		6				10
19.	Процессы и домены приложения. Валидация модели.	2		6				10
	Курсовая работа							**10**
	Промежуточная аттестация	**18,15**						
	Итого	**68**		**84**				**189,85**

6.2 Программа дисциплины, структурированная по темам / разделам

6.2.1 Содержание лекционного курса

№ п/п	Наименование темы (раздела) дисциплины	Содержание лекционного занятия
1.	Объекты ООП.	Классы и объекты. Конструкторы, инициализаторы и деконструкторы.
2.	Пространства имен.	Пространства имен. Глобальные пространства имен.
3.	Модификаторы.	Модификаторы доступа. Свойства. Перегрузка методов. Поля и структуры для чтения.
4.	Наследование.	Наследование. Преобразование типов. Перегрузка операций преобразования типов. Виртуальные методы и свойства.
5.	Абстрактные классы.	Абстрактные классы. Обобщенные типы. Ограничения обобщений.
6.	Обработка исключений.	Исключения и фильтры исключений. Типы исключений. Генерация исключения.
7.	Делегаты, события и лямбды.	Делегаты. Анонимные методы. Лямбды. События. Ковариантность и контравариантность делегатов. Замыкания.
8.	Интерфейсы.	Определение интерфейсов. Применение интерфейсов. Явная реализация интерфейсов. Реализация интерфейсов в базовых и производных классах.

		Наследование интерфейсов. Интерфейсы в обобщениях. Копирование объектов.
9.	Дополнительные возможности ООП в C#.	Переменные-ссылки и возвращение ссылки. Методы расширения. Частичные классы и методы. Анонимные типы. Кортежи.
10.	Коллекции	Список. Двухсвязный список. Очередь. Стек. Словарь. Итераторы.
11.	Работа со строками	Операции со строками. Форматирование и интерполяция строк. Регулярные выражения. Работа с датами и временем.
12.	Дополнительные классы и структуры .NET	Отложенная инициализация. Математические вычисления. Преобразование типов. Индексы и диапазоны.
13.	Многопоточность	Введение в многопоточность. Создание потоков. Потоки с параметрами. Синхронизация потоков. Мониторы. Мьютексы. Семафоры. Таймеры. Задачи и класс Task. Задачи продолжения. Класс Parallel. Отмена задач и параллельных операций.
14.	Асинхронное программирование	Асинхронные методы. Возвращение результата из асинхронного метода. Последовательный и параллельный вызов асинхронных операций.
15.	LINQ	Фильтрация выборки и проекция. Сортировка. Объединение, пересечение и разность коллекций. Агрегатные операции. Группировка. Соединение коллекций. Отложенное и немедленное выполнение LINQ. Делегаты и анонимные методы в запросах LINQ.
16.	Рефлексия. Dynamic Language Runtime. Сборка мусора, управление памятью и указатели.	Введение в рефлексию. Применение рефлексии и исследование типов. Динамическая загрузка сборок и позднее связывание. Сборщик мусора в C#. Финализируемые объекты. Указатели. Указатели на структуры, члены классов и массивы.
17.	Работа с потоками и файловой системой	Работа с дисками. Работа с каталогами. Работа с файлами. Чтение и запись файла. Чтение и запись текстовых файлов. Бинарные файлы. Бинарная сериализация. Архивация и сжатие файлов.
18.	Сериализация.	Сериализация в JSON. XML-Документы. Изменение XML-документа. Создание Xml-документа.
19.	Процессы и домены приложения. Валидация модели.	Процессы. Домены приложений. Основы валидации модели. Атрибуты валидации.

6.2.2 Содержание практических занятий

№ п/п	Наименование темы (раздела) дисциплины	Содержание практического занятия
1.	Объекты ООП.	Создание объекта класса. Обращение к функциональности класса. Создание конструкторов. Класс Program и метод Main. Программы верхнего уровня.
2.	Пространства имен.	Подключение пространств имен по умолчанию.

		Создание библиотеки классов.
3.	Модификаторы.	Модификаторы в рамках текущего проекта. Модификаторы в рамках сборок. Определение свойств. Перегрузка методов. Перегрузка методов.
4.	Наследование.	Доступ к членам базового класса из класса-наследника. Конструкторы в производных классах.
5.	Абстрактные классы.	Класс System. Object и его методы. Обобщенные типы. Ограничения обобщений. Наследование обобщенных типов.
6.	Обработка исключений.	Конструкция try. catch. finally. Блок catch и фильтры исключений. Класс Exception. Поиск блока catch при обработке исключений. Оператор throw.
7.	Делегаты, события и лямбды.	Делегаты. Лямбды. События. ДелегатыAction, PredicateиFunc. Замыкания.
8.	Интерфейсы.	Явная реализация интерфейсов. Реализация интерфейсов в базовых и производных классах. Наследование интерфейсов. Интерфейсы в обобщениях. Интерфейс ICloneable. Интерфейс IComparable.
9.	Дополнительные возможности ООП в C#.	Null и ссылочные типы. Null и значимые типы. Проверка на null, операторы? и ??. Частичные классы и методы. Анонимные типы. Кортежи. Patternmatching. Records.
10.	Коллекции	ArrayList. Список List<T>. Двухсвязный список LinkedList<T>. Очередь Queue<T>. СтекStack<T>. СловарьDictionary<T, V>. КлассObservableCollection. ИнтерфейсыIEnumerableиIEnumerator. Итераторы и оператор yield.
11.	Работа со строками	Строки и класс System.String. Операции со строками. Форматирование и интерполяция строк. Класс StringBuilder. Работа с датами и временем.
12.	Дополнительные классы и структуры .NET	Отложенная инициализация и тип Lazy. Математические вычисления и класс Math. Преобразование типов и класс Convert. Класс Array и массивы. Span.
13.	Многопоточность	Класс Thread. ДелегатThreadStart. ПотокиспараметрамииParameterizedThreadStart. Синхронизация потоков. Класс AutoResetEvent. Семафоры. Таймеры. Задачи и класс Task. Работа с классом Task. Класс Parallel. CancellationToken.
14.	Асинхронное программирование	Асинхронные методы, async и await. Обработка ошибок в асинхронных методах. Отмена асинхронных операций. Асинхронные стримы.
15.	LINQ	Основы LINQ. Фильтрация выборки и проекция. Сортировка. МетодыSkipиTake. МетодJoin, GroupJoinиZip. Методы All и Any. Делегаты и анонимные методы в запросах LINQ.
16.	Рефлексия. Dynamic Language Runtime. Сборка мусора, управление памятью и указатели.	Класс System.Type. Атрибуты в .NET. DLR в C#. Ключевоеслово dynamic. DynamicObjectиExpandoObject. Использование IronPython в .NET. Метод Dispose. Конструкция using. Указатели. Указатели на структуры, члены классов и массивы.

17.	Работа с потоками и файловой системой	Классы File и FileInfo. FileStream. StreamReaderиStreamWriter. BinaryWriterиBinaryReader. BinaryFormatter. Архивация и сжатие файлов.
18.	Сериализация.	JsonSerializer. Работа с XML с помощью System.Xml. XPath. Linq to Xml. Создание Xml-документа. Выборка элементов в LINQ to XML. XmlSerializer.
19.	Процессы и домены приложения. Валидация модели.	Процессы. AssemblyLoadContext и динамическая загрузка и выгрузка сборок. Создание своих атрибутов валидации. Самовалидация модели.

6.2.3 Содержание самостоятельной работы

№ п/п	Наименование темы (раздела) дисциплины	Содержание самостоятельной работы
1.	Объекты ООП.	Классы и объекты. Конструкторы, инициализаторы и деконструкторы. Класс Program и метод Main. Программы верхнего уровня.
2.	Пространства имен.	Пространства имен. Глобальные пространства имен. Подключение пространств имен по умолчанию. Создание библиотеки классов.
3.	Модификаторы.	Модификаторы доступа. Свойства. Перегрузка методов. Статические члены и модификатор static. Поля и структуры для чтения. Перегрузка операторов. Индексаторы.
4.	Наследование.	Наследование. Преобразование типов. Перегрузка операций преобразования типов. Виртуальные методы и свойства. Сокрытие методов. Различие переопределения и сокрытия методов.
5.	Абстрактные классы.	Абстрактные классы. Класс System.Object и его методы. Обобщенные типы. Ограничения обобщений. Наследование обобщенных типов.
6.	Обработка исключений.	Конструкция try..catch..finally. Блок catch и фильтры исключений. Типы исключений. Класс Exception. Создание классов исключений. Поиск блока catch при обработке исключений. Генерация исключения и оператор throw.
7.	Делегаты, события и лямбды.	Делегаты. Применение делегатов. Анонимные методы. Лямбды. События. Ковариантность и контравариантность делегатов. Делегаты Action, Predicate и Func. Замыкания.
8.	Интерфейсы.	Определение интерфейсов. Применение интерфейсов. Явная реализация интерфейсов. Реализация интерфейсов в базовых и производных классах. Наследование интерфейсов. Интерфейсы в обобщениях. Копирование объектов. Интерфейс ICloneable. Сортировка объектов. Интерфейс IComparable. Ковариантность и контравариантность обобщенных интерфейсов
9.	Дополнительные возможности ООП в C#.	Null и ссылочные типы. Null и значимые типы. Проверка на null, операторы ?. и ??. Переменные-ссылки и возвращение ссылки. Методы расширения.

		Частичные классы и методы. Анонимные типы. Кортежи. Patternmatching. Records.
10	Коллекции	ArrayList. Список List<T>. Двухсвязный список LinkedList<T>. Очередь Queue<T>. СтекStack<T>. СловарьDictionary<T, V>. КлассObservableCollection. ИнтерфейсыIEnumerableиIEnumerator. Итераторы и оператор yield.
11	Работа со строками	Работа со строками. Строки и класс System.String. Операции со строками. Форматирование и интерполяция строк. Класс StringBuilder. Регулярные выражения. Работа с датами и временем.
12	Дополнительные классы и структуры .NET	Отложенная инициализация и тип Lazy. Математические вычисления и класс Math. Преобразование типов и класс Convert. Класс Array и массивы. Span. Индексы и диапазоны.
13	Многопоточность	Введение в многопоточность. Класс Thread. Создание потоков. Делегат ThreadStart. Потоки с параметрами и ParameterizedThreadStart. Синхронизация потоков. Мониторы. Класс AutoResetEvent. Мьютексы. Семафоры. Таймеры. Задачи и класс Task. Работа с классом Task. Задачи продолжения. Класс Parallel. Отмена задач и параллельных операций. CancellationToken.
14	Асинхронное программирование	Асинхронные методы, async и await. Возвращение результата из асинхронного метода. Последовательный и параллельный вызов асинхронных операций. Обработка ошибок в асинхронных методах. Отмена асинхронных операций. Асинхронные стримы.
15	LINQ	Основы LINQ. Фильтрация выборки и проекция. Сортировка. Объединение, пересечение и разность коллекций. Агрегатные операции. Методы Skip и Take. Группировка. Соединение коллекций. Метод Join, GroupJoin и Zip. Методы All и Any. Отложенное и немедленное выполнение LINQ. Делегаты и анонимные методы в запросах LINQ.
16	Рефлексия. Dynamic Language Runtime. Сборка мусора, управление памятью и указатели.	Введение в рефлексию. Класс System.Type. Применение рефлексии и исследование типов. Динамическая загрузка сборок и позднее связывание. Атрибуты в .NET. DLR в C#. Ключевое слово dynamic. DynamicObject и ExpandoObject. Использование IronPython в .NET. Сборщик мусора в C#. Финализируемые объекты. Метод Dispose. Конструкция using. Указатели. Указатели на структуры, члены классов и массивы.
17	Работа с потоками и файловой системой	Работа с дисками. Работа с каталогами. Работа с файлами. Классы File и FileInfo. FileStream. Чтение и запись файла. Чтение и запись текстовых файлов. StreamReader и StreamWriter. Бинарные файлы. BinaryWriter и BinaryReader. Бинарная сериализация. BinaryFormatter. Архивация и сжатие файлов.

18	Сериализация.	Сериализация в JSON. JsonSerializer. XML-Документы. Работа с XML с помощью System.Xml. Изменение XML-документа. XPath. LinqtoXml. Создание Xml-документа. Выборка элементов в LINQ to XML. Изменение документа в LINQ to XML. Сериализация в XML. XmlSerializer.
19	Процессы и домены приложения. Валидация модели.	Процессы. Домены приложений. AssemblyLoadContext и динамическая загрузка и выгрузка сборок. Основы валидации модели. Атрибуты валидации. Создание своих атрибутов валидации. Самовалидация модели.

7. Текущий контроль по дисциплине (модулю) в рамках учебных занятий

В рамках текущего контроля преподаватель самостоятельно может проводить следующие мероприятия:

№ п/п	Контролируемые разделы (темы)	Наименование оценочного средства
1.	Объекты ООП.	Опрос, проблемно-аналитическое задание, тестирование.
2.	Пространства имен.	Опрос, проблемно-аналитическое задание, тестирование.
3.	Модификаторы.	Опрос, проблемно-аналитическое задание, тестирование.
4.	Наследование.	Опрос, проблемно-аналитическое задание, тестирование.
5.	Абстрактные классы.	Опрос, проблемно-аналитическое задание, тестирование.
6.	Обработка исключений.	Опрос, проблемно-аналитическое задание, тестирование.
7.	Делегаты, события и лямбды.	Опрос, проблемно-аналитическое задание, тестирование.
8.	Интерфейсы.	Опрос, проблемно-аналитическое задание, тестирование.
9.	Дополнительные возможности ООП в C#.	Опрос, проблемно-аналитическое задание, тестирование.
10.	Коллекции	Опрос, проблемно-аналитическое задание, тестирование.
11.	Работа со строками	Опрос, проблемно-аналитическое задание, тестирование.
12.	Дополнительные классы и структуры .NET	Опрос, проблемно-аналитическое задание, тестирование.
13.	Многопоточность	Опрос, проблемно-аналитическое задание, тестирование.
14.	Асинхронное программирование	Опрос, проблемно-аналитическое задание, тестирование.
15.	LINQ	Опрос, проблемно-аналитическое задание, тестирование.
16.	Рефлексия. Dynamic Language Runtime. Сборка мусора, управление	Опрос, проблемно-аналитическое задание, тестирование.

	памятью и указатели.	
17.	Работа с потоками и файловой системой	Опрос, проблемно-аналитическое задание, тестирование.
18.	Сериализация.	Опрос, проблемно-аналитическое задание, тестирование.
19.	Процессы и домены приложения. Валидация модели.	Опрос, проблемно-аналитическое задание, тестирование.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

8.1. Основная учебная литература

1. Объектно-ориентированное программирование: лабораторный практикум /. — Ставрополь: Северо-Кавказский федеральный университет, 2018. — 111 с. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: https://www.iprbookshop.ru/92712.html

2. Объектно-ориентированное программирование на C++: учебник / И.В. Баранова [и др.]. — Красноярск: Сибирский федеральный университет, 2019. — 288 с. — ISBN 978-5-7638-4034-6. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: https://www.iprbookshop.ru/100067.html

8.2. Дополнительная учебная литература:

1. Лебедева Т.Н. Теория и практика объектно-ориентированного программирования: учебное пособие / Лебедева Т.Н. — Челябинск, Саратов: Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. — 221 с. — ISBN 978-5-4486-0663-2. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: https://www.iprbookshop.ru/81498.html

2. Литвиненко В.А. Основы объектно-ориентированного программирования задач на графах: учебное пособие / Литвиненко В.А. — Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2019. — 133 с. — ISBN 978-5-9275-3472-2. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: https://www.iprbookshop.ru/107969.html

3. Мейер Б. Объектно-ориентированное программирование и программная инженерия / Мейер Б. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 285 с. — ISBN 978-5-4486-0513-0. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: https://www.iprbookshop.ru/79706.html

8.3. Периодические издания

1. Журнал «Математическое моделирование и численные методы». Математическое моделирование и численные методы (bmstu.ru)
2. Вестник Московского Университета. Математика, Механика (msu.su)
3. Дискретнаяматематика. Discrete Mathematics and Applications. (mathnet.ru)

9. Перечень ресурсов информационно-телекоммуникационной сети "Интернет" (далее - сеть "Интернет"), необходимых для освоения дисциплины (модуля)

1. Федеральный портал «Российское образование». http://www.edu.ru/
2. Электронно-библиотечная система «Научная электронная библиотека eLIBRARY.RU» https://www.elibrary.ru /
3. Электронно-библиотечная система ЛАНЬ https://e.lanbook.com/
4. Электронно-библиотечная система IPR BOOKS https://www.iprbookshop.ru

5. https://www.rsl.ru - Российская Государственная Библиотека (ресурсы открытого доступа)

6. https://link.springer.com - Международная реферативная база данных научных изданий Springerlink (ресурсы открытого доступа)

7. https://zbmath.org - Международная реферативная база данных научных изданий zbMATH (ресурсы открытого доступа)

8. https://openedu.ru - «Национальная платформа открытого образования» (ресурсы открытого доступа)

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

Успешное освоение данного курса базируется на рациональном сочетании нескольких видов учебной деятельности – лекций, семинарских занятий, самостоятельной работы. При этом самостоятельную работу следует рассматривать одним из главных звеньев полноценного высшего образования, на которую отводится значительная часть учебного времени.

Самостоятельная работа студентов складывается из следующих составляющих:

1. работа с основной и дополнительной литературой, с материалами интернета и конспектами лекций;
2. внеаудиторная подготовка к контрольным работам, выполнение докладов, рефератов и курсовых работ;
3. выполнение самостоятельных практических работ;
4. подготовка к экзаменам (зачетам) непосредственно перед ними.

Для правильной организации работы необходимо учитывать порядок изучения разделов курса, находящихся в строгой логической последовательности. Поэтому хорошее усвоение одной части дисциплины является предпосылкой для успешного перехода к следующей. Задания, проблемные вопросы, предложенные для изучения дисциплины, в том числе и для самостоятельного выполнения, носят междисциплинарный характер и базируются, прежде всего, на причинно-следственных связях между компонентами окружающего нас мира. В течение семестра, необходимо подготовить рефераты (проекты) с использованием рекомендуемой основной и дополнительной литературы и сдать рефераты для проверки преподавателю. Важным составляющим в изучении данного курса является решение ситуационных задач и работа над проблемно-аналитическими заданиями, что предполагает знание соответствующей научной терминологии и т.д.

Для лучшего запоминания материала целесообразно использовать индивидуальные особенности и разные виды памяти: зрительную, слуховую, ассоциативную. Успешному запоминанию также способствует приведение ярких свидетельств и наглядных примеров. Учебный материал должен постоянно повторяться и закрепляться.

При выполнении докладов, творческих, информационных, исследовательских проектов особое внимание следует обращать на подбор источников информации и методику работы с ними.

Для успешной сдачи экзамена (зачета) рекомендуется соблюдать следующие правила:

1. Подготовка к экзамену (зачету) должна проводиться систематически, в течение всего семестра.
2. Интенсивная подготовка должна начаться не позднее, чем за месяц до экзамена.
3. Время непосредственно перед экзаменом (зачетом) лучше использовать таким образом, чтобы оставить последний день свободным для повторения курса в целом, для систематизации материала и доработки отдельных вопросов.

На экзамене высокую оценку получают студенты, использующие данные, полученные в процессе выполнения самостоятельных работ, а также использующие собственные выводы на основе изученного материала.

Учитывая значительный объем теоретического материала, студентам рекомендуется регулярное посещение и подробное конспектирование лекций.

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

1. Microsoft Windows Server;
2. Семейство ОС Microsoft Windows;
3. Libre Office свободно распространяемый офисный пакет с открытым исходным кодом;
4. Информационно-справочная система: Система КонсультантПлюс (КонсультантПлюс);
5. Информационно-правовое обеспечение Гарант: Электронный периодический справочник «Система ГАРАНТ» (Система ГАРАНТ);

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

12.1. Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения.

Специализированная мебель:

Комплект учебной мебели (стол, стул) по количеству обучающихся; комплект мебели для преподавателя; доска (маркерная).

Технические средства обучения:

Компьютер в сборе для преподавателя, колонки, проектор, экран.

Перечень лицензионного программного обеспечения, в том числе отечественного производства: Windows 10, КонсультантПлюс, Kaspersky Endpoint Security.

Перечень свободно распространяемого программного обеспечения:

Yandex Browser, пакет LibreOffice, МТС Линк, Gimp, FreeCAD.

1) IDE Visual Studio Community (нагрузка «Разработка классических приложений на C++» с компонентом «Поддержка C++/CLI»; поддержка MFC)

2) СУБД MySQL (клиент-серверная)

3) Ramus Modelio

4) Cisco Packet Tracer (версии 7.x и 8.x)

5) Oracle Virtual Box

6) Adobe Reader

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду ММУ.

12.2. Помещение для самостоятельной работы обучающихся.

Специализированная мебель:

Комплект учебной мебели (стол, стул) по количеству обучающихся; комплект мебели для преподавателя; доска (маркерная).

Технические средства обучения:

Компьютер в сборе для преподавателя; компьютеры в сборе для обучающихся; колонки; проектор, экран.

Перечень лицензионного программного обеспечения, в том числе отечественного производства: Windows 10, КонсультантПлюс, Kaspersky Endpoint Security.

Перечень свободно распространяемого программного обеспечения:

Adobe Reader, Yandex Browser, пакет LibreOffice, МТС Линк, Gimp, FreeCAD.

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду ММУ.

13.Образовательные технологии, используемые при освоении дисциплины

Для освоения дисциплины используются как традиционные формы занятий – лекции (типы лекций – установочная, вводная, текущая, заключительная, обзорная; виды лекций – проблемная, визуальная, лекция конференция, лекция консультация); и семинарские (практические) занятия, так и активные и интерактивные формы занятий - деловые и ролевые игры, решение ситуационных задач и разбор конкретных ситуаций.

На учебных занятиях используются технические средства обучения мультимедийной аудитории: компьютер, монитор, колонки, настенный экран, проектор, микрофон, пакет программ Microsoft Office для демонстрации презентаций и медиафайлов, видеопроектор для демонстрации слайдов, видеосюжетов и др. Тестирование обучаемых может осуществляться с использованием компьютерного оборудования университета.

13.1. В освоении учебной дисциплины используются следующие традиционные образовательные технологии:

- чтение проблемно-информационных лекций с использованием доски и видеоматериалов;
- семинарские занятия для обсуждения, дискуссий и обмена мнениями;
- контрольные опросы;
- консультации;
- самостоятельная работа студентов с учебной литературой и первоисточниками;
- подготовка и обсуждение рефератов (проектов), презентаций (научно-исследовательская работа);
- тестирование по основным темам дисциплины.

13.2. Активные и интерактивные методы и формы обучения

Из перечня видов: («*мозговой штурм», анализ НПА, анализ проблемных ситуаций, анализ конкретных ситуаций, инциденты, имитация коллективной профессиональной деятельности, разыгрывание ролей, творческая работа, связанная с освоением дисциплины, ролевая игра, круглый стол, диспут, беседа, дискуссия, мини-конференция и др.*) используются следующие:

- *диспут*
- *анализ проблемных, творческих заданий, ситуационных задач*
- *ролевая игра;*
- *круглый стол;*
- *мини-конференция*
- *дискуссия*
- *беседа.*

13.3. Особенности обучения инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ)

При организации обучения по дисциплине учитываются особенности организации взаимодействия с инвалидами и лицами с ограниченными возможностями здоровья (далее – инвалиды и лица с ОВЗ) с целью обеспечения их прав. При обучении учитываются особенности их психофизического развития, индивидуальные возможности и при необходимости обеспечивается коррекция нарушений развития и социальная адаптация указанных лиц.

Выбор методов обучения определяется содержанием обучения, уровнем методического и материально-технического обеспечения, особенностями восприятия учебной информации студентов-инвалидов и студентов с ограниченными возможностями здоровья и т.д. В образовательном процессе используются социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в

установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе.

При обучении лиц с ограниченными возможностями здоровья электронное обучение и дистанционные образовательные технологии предусматривают возможность приема-передачи информации в доступных для них формах.

Обучающиеся из числа лиц с ограниченными возможностями здоровья обеспечены печатными и электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

Приложение

Автономная некоммерческая организация высшего образования
«МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ УНИВЕРСИТЕТ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

Объектно-ориентированное программирование

Направление подготовки	Информационные системы и технологии
Код	09.03.02
Направленность (профиль)	Проектирование, разработка и сопровождение информационных систем
Квалификация выпускника	бакалавр

Москва
2023

1. Перечень кодов компетенций, формируемых дисциплиной в процессе освоения образовательной программы

Группа компетенций	Категория компетенций	Код
Профессиональные	-	ПК-5

2. Компетенции и индикаторы их достижения

Код компетенции	Формулировка компетенции	Индикаторы достижения компетенции
ПК-5	Способен выполнять работы по созданию (модификации) и сопровождению ИС.	**ПК-5.1.** Типовое проектирование информационных систем, а также различных моделей информационных систем и проектных спецификаций; Программные прототипы решения прикладных задач. **ПК-5.2.** Разработка ИС с учетом требований заказчика, на основе стандартов к проектированию информационных систем. Модификация существующих ИС для улучшения их функциональности и производительности. **ПК-5.3.** Способность разрабатывать мобильные приложения и работать с Интернет вещами **ПК-5.4.** Знать и уметь работать с технологиями искусственного интеллекта и инструментальными средствами разработки интеллектуальных программных систем. **ПК-5.5.** Верификация структуры программного кода ИС относительно архитектуры ИС. **ПК-5.6.** Создание пользовательские интерфейсы с учетом UX/UI принципов для повышения удобства использования ИС. **ПК-5.7.** Осуществляет поиск, анализ, программную реализацию математических моделей и алгоритмов интеллектуальной обработки данных.

3. Описание планируемых результатов обучения по дисциплине

3.1. Описание планируемых результатов обучения по дисциплине

Планируемые результаты обучения по дисциплине представлены дескрипторами (знания, умения, навыки).

Дескрипторы по дисциплине	Знать	Уметь	Владеть
Код компетенции	**ПК-5**		

	-современные методы разработки и реализации алгоритмов на базе языков программирования и пакетов прикладных программ; - основы современных систем управления базами данных, теории баз данных; - формальные методы, технологии и инструменты разработки программного обеспечения и баз данных; - основы программирования, современные объектно-ориентированные языки программирования; современные структурные языки программирования, языки современных бизнес-приложений; - современные методики тестирования разрабатываемых ИС: инструменты и методы модульного тестирования, инструменты и методы тестирования нефункциональных и функциональных характеристик ИС.	-разрабатывать алгоритмы и программы на базе языков программирования и пакетов прикладных программ, пригодные для практического применения; - кодировать на языках программирования; - тестировать результаты кодирования;	- приемами разработки алгоритмов и программ на базе языков программирования и пакетов прикладных программ, пригодных для практического применения.

3.2. Критерии оценки результатов обучения по дисциплине

Шкала оценивания	**Индикаторы достижения**	**Показатели оценивания результатов обучения**

ОТЛИЧНО/ЗАЧТЕНО	Знает:	- студент глубоко и всесторонне усвоил материал, уверенно, логично, последовательно и грамотно его излагает, опираясь на знания основной и дополнительной литературы, - на основе системных научных знаний делает квалифицированные выводы и обобщения, свободно оперирует категориями и понятиями.
	Умеет:	- студент умеет самостоятельно и правильно решать учебно-профессиональные задачи или задания, уверенно, логично, последовательно и аргументировано излагать свое решение, используя научные понятия, ссылаясь на нормативную базу.
	Владеет:	- студент владеет рациональными методами (с использованием рациональных методик) решения сложных профессиональных задач, представленных деловыми играми, кейсами и т.д.; При решении продемонстрировал навыки - выделения главного, - связкой теоретических положений с требованиями руководящих документов, - изложения мыслей в логической последовательности, - самостоятельного анализа факты, событий, явлений, процессов в их взаимосвязи и диалектическом развитии.
ХОРОШО/ЗАЧТЕНО	Знает:	- студент твердо усвоил материал, достаточно грамотно его излагает, опираясь на знания основной и дополнительной литературы, - затрудняется в формулировании квалифицированных выводов и обобщений, оперирует категориями и понятиями, но не всегда правильно их верифицирует.
	Умеет:	- студент умеет самостоятельно и в основном правильно решать учебно-профессиональные задачи или задания, уверенно, логично, последовательно и аргументировано излагать свое решение, не в полной мере используя научные понятия и ссылки на нормативную базу.
	Владеет:	- студент в целом владеет рациональными методами решения сложных профессиональных задач, представленных деловыми играми, кейсами и т.д.; При решении смог продемонстрировать достаточность, но не глубинность навыков - выделения главного, - изложения мыслей в логической последовательности. - связки теоретических положений с требованиями руководящих документов, - самостоятельного анализа факты, событий, явлений, процессов в их взаимосвязи и диалектическом развитии.
УДОВЛЕТВОРИТЕЛЬНО/ ЗАЧТЕНО	Знает:	- студент ориентируется в материале, однако затрудняется в его изложении; - показывает недостаточность знаний основной и дополнительной литературы; - слабо аргументирует научные положения; - практически не способен сформулировать выводы и обобщения; - частично владеет системой понятий.

	Умеет:	- студент в основном умеет решить учебно-профессиональную задачу или задание, но допускает ошибки, слабо аргументирует свое решение, недостаточно использует научные понятия и руководящие документы.
	Владеет:	- студент владеет некоторыми рациональными методами решения сложных профессиональных задач, представленных деловыми играми, кейсами и т.д.; При решении продемонстрировал недостаточность навыков - выделения главного, - изложения мыслей в логической последовательности. - связки теоретических положений с требованиями руководящих документов, - самостоятельного анализа факты, событий, явлений, процессов в их взаимосвязи и диалектическом развитии.
Компетенция не достигнута		
НЕУДОВЛЕТВОРИТЕЛЬНО/ НЕ ЗАЧТЕНО	Знает:	- студент не усвоил значительной части материала; - не может аргументировать научные положения; - не формулирует квалифицированных выводов и обобщений; - не владеет системой понятий.
	Умеет:	студент не показал умение решать учебно-профессиональную задачу или задание.
	Владеет:	не выполнены требования, предъявляемые к навыкам, оцениваемым “удовлетворительно”.

При ответе на вопросы в рамках прохождения промежуточной аттестации (зачет/ зачет с оценкой/ экзамен) допускается вольная формулировка ответа, по смыслу раскрывающая содержание ответа, указанного в фонде оценочных средств, в качестве верного ответа.

При подготовке ответа в рамках прохождения промежуточной аттестации (зачет/ зачет с оценкой/ экзамен) обучающимся разрешается использовать калькулятор и справочные таблицы.

4. Типовые контрольные задания (закрытого, открытого и иного типа) для проведения промежуточной аттестации, необходимые для оценки достижения компетенции, соотнесенной с результатами обучения по дисциплине

3 СЕМЕСТР

ПК-5

1.**Что такое объектно-ориентированное программирование (ООП)?**

a) Методология программирования, основанная на объектах и классах.

b) Методология программирования, основанная на функциях.

c) Методология программирования, основанная на массивах.

d) Методология программирования, основанная на строках.

Ответ: a) Методология программирования, основанная на объектах и классах.

2. **Что такое класс в C++?**

a) Шаблон для создания объектов.
b) Функция, которая выполняет определенные действия.
c) Переменная, которая хранит данные.
d) Массив, который хранит объекты.
Ответ: a) Шаблон для создания объектов.

3. Напишите, ч**то такое объект в C++.**
Ответ: объект — это экземпляр класса.

4. Назовите основные принципы ООП, и поясните, что каждый из них означает.
Ответ: Инкапсуляция: Скрытие деталей реализации от пользователя. Объект предоставляет интерфейс для взаимодействия, не раскрывая внутреннюю структуру.
1.**Наследование**: Возможность создания нового класса на основе существующего. Производный класс наследует свойства и методы базового класса и может их расширять или изменять.
2.**Полиморфизм**: Возможность использования одного и того же интерфейса для работы с разными типами данных. Полиморфизм позволяет писать более гибкий и расширяемый код.
3.**Абстракция**: Выделение только важных характеристик объекта и игнорирование несущественных деталей. Абстракция помогает создавать модели реальных объектов, фокусируясь на их основных свойствах.

5. Какой модификатор доступа по умолчанию для членов класса в C++?
a) private
b) public
c) protected
d) friend
Ответ: a) private

6. Напишите, что такое конструктор в C++.
Ответ: Конструктор — это специальный метод класса, который автоматически вызывается при создании объекта. Его основная цель — инициализировать поля объекта и выполнять другие необходимые действия для подготовки объекта к использованию. Конструктор имеет то же имя, что и класс, и не имеет возвращаемого типа.

7. Как называется конструктор, который не принимает аргументов?
a) Конструктор по умолчанию.
b) Конструктор копирования.
c) Конструктор перемещения.
d) Конструктор инициализации.
Ответ: a) Конструктор по умолчанию

8. Что будет выведено на экран при выполнении этого кода?
```
class MyClass {
public:
   MyClass() {
      std::cout << "Constructor called" << std::endl;
   }
   ~MyClass() {
      std::cout << "Destructor called" << std::endl;
   }
};
```

```
int main() {
   MyClass obj;
   return 0;
}
```
Ответ:
Constructor called
Destructor called

9. Что будет выведено на экран при выполнении этого кода?
```
class MyClass {
public:
   MyClass() {
      std::cout << "Default constructor called" << std::endl;
   }
   MyClass(const MyClass& other) {
      std::cout << "Copy constructor called" << std::endl;
   }
};

int main() {
   MyClass obj1;
   MyClass obj2 = obj1;
   return 0;
}
```
Ответ:
Default constructor called
Copy constructor called

10. Что будет выведено на экран при выполнении этого кода?

```
class MyClass {
private:
   int data;
public:
   MyClass(int d) : data(d) {}
   friend void showData(MyClass obj);
};

void showData(MyClass obj) {
   std::cout << "Data: " << obj.data << std::endl;
}

int main() {
   MyClass obj(42);
   showData(obj);
   return 0;
}
```

a) Data: 0
b) Data: 42

c) Data: -1
d) Ничего не будет выведено

Ответ: b) Data: 42

11. Что такое шаблон функции в C++?

Варианты ответа:

a) Функция, которая работает только с одним типом данных.

b) Функция, которая может работать с разными типами данных.

c) Функция, которая вызывается только в базовом классе.

d) Функция, которая вызывается только в производном классе.

Правильный ответ:

b) Функция, которая может работать с разными типами данных.

12. Что такое шаблон класса в C++?

Варианты ответа:

a) Определение класса, которое работает только с одним типом данных.

b) Обобщенное определение класса, которое может работать с разными типами данных.

c) Определение класса, которое работает только с числовыми типами данных.

d) Определение класса, которое работает только со строковыми типами данных.

Правильный ответ:

b) Обобщенное определение класса, которое может работать с разными типами данных.

13. Что такое приватные методы в C++?

Варианты ответа:

a) Методы, которые доступны только в базовом классе.

b) Методы, которые доступны только внутри класса.

c) Методы, которые доступны только в производном классе.

d) Методы, которые доступны только в дружественных классах.

Правильный ответ:

b) Методы, которые доступны только внутри класса.

14. Что такое защищенные методы в C++?

Варианты ответа:

a) Методы, которые доступны только в базовом классе.

b) Методы, которые доступны только в производном классе.

c) Методы, которые доступны только внутри класса и его производных классов.

d) Методы, которые доступны только в дружественных классах.

Правильный ответ:

c) Методы, которые доступны только внутри класса и его производных классов.

15. Что такое дружественные функции в C++?

Варианты ответа:

a) Функции, которые вызываются только в базовом классе.

b) Функции, которые не имеют доступа к закрытым членам класса.

c) Функции, которые имеют доступ к закрытым членам класса.

d) Функции, которые вызываются только в производном классе.

Правильный ответ:

c) Функции, которые имеют доступ к закрытым членам класса.

16. Что такое статические методы в C++?

Варианты ответа:
a) Методы, которые могут быть вызваны только в базовом классе.
b) Методы, которые принадлежат классу, а не объекту.
c) Методы, которые принадлежат объекту, а не классу.
d) Методы, которые могут быть вызваны только в производном классе.
Правильный ответ:
b) Методы, которые принадлежат классу, а не объекту.

17. Что такое константные методы в C++?
Варианты ответа:
a) Методы, которые могут быть вызваны только в базовом классе.
b) Методы, которые изменяют состояние объекта.
c) Методы, которые не изменяют состояние объекта.
d) Методы, которые могут быть вызваны только в производном классе.
Правильный ответ:
c) Методы, которые не изменяют состояние объекта.

18. Как объявить шаблон функции с несколькими параметрами типа в C++?
Варианты ответа:
a) template <typename T1, T2> T1 function(T1 a, T2 b);
b) template <typename T1, typename T2> T1 function(T1 a, T2 b);
c) template <typename T1, typename T2> T1 function(T2 a, T1 b);
d) template <typename T1, T2> T1 function(T2 a, T1 b);
Правильный ответ:
b) template <typename T1, typename T2> T1 function(T1 a, T2 b);

19. Как объявить шаблон класса с несколькими параметрами типа в C++?
Варианты ответа:
a) template <typename T1, T2> class MyClass;
b) template <typename T1, typename T2> class MyClass;
c) template <typename T1, typename T2> MyClass;
d) template <typename T1, T2> MyClass;
Правильный ответ:
b) template <typename T1, typename T2> class MyClass;

20. Что такое дружественный класс в C++?
Варианты ответа:
a) Класс, который не имеет доступа к закрытым членам другого класса.
b) Класс, который содержит только статические методы.
c) Класс, который имеет доступ к закрытым членам другого класса.
d) Класс, который содержит только константы.
Правильный ответ:
c) Класс, который имеет доступ к закрытым членам другого класса.

Задания открытого типа:

1. Что такое компонентное программирование?
2. Дайте определения: делегаты, события и лямбды.
3. Назовите объекты ООП.

№	Вопрос	Ответ
1	Что такое компонентное программирование?	Компонентное программирование — это подход к разработке программного обеспечения, основанный на создании и использовании независимых модулей (компонентов), которые можно повторно использовать, легко модифицировать и интегрировать в системы.
2	Дайте определения: делегаты, события и лямбды.	**Делегаты** — типы, представляющие ссылки на методы, которые можно вызвать, позволяя передавать методы как параметры. **События** — механизм уведомления об изменении состояния объекта, основанный на делегатах, позволяющий подписываться и реагировать на изменения. **Лямбды** — анонимные функции, используемые для упрощения записи методов, чаще всего применяемые в делегатах или LINQ.
3	Назовите объекты ООП.	Объекты объектно-ориентированного программирования: классы, объекты, атрибуты (свойства), методы, наследование, полиморфизм, инкапсуляция, абстракция

4 СЕМЕСТР

ПК-5

1. Какой модификатор доступа используется для наследования с доступом к защищенным членам базового класса?

Варианты ответа:
a) public
b) protected
c) private
d) friend
Правильное решение:
b) protected

2. Что такое виртуальная функция в C++?

Варианты ответа:
a) Функция, которая может быть переопределена в производном классе.
b) Функция, которая не может быть переопределена в производном классе.
c) Функция, которая вызывается только в базовом классе.
d) Функция, которая вызывается только в производном классе.
Правильное решение:
a) Функция, которая может быть переопределена в производном классе.

3. Что такое абстрактный класс в C++?

Варианты ответа:
a) Класс, который содержит хотя бы одну чистую виртуальную функцию.

b) Класс, который не содержит виртуальных функций.
c) Класс, который содержит только статические методы.
d) Класс, который содержит только константы.
Правильное решение:
a) Класс, который содержит хотя бы одну чистую виртуальную функцию.

4. Создайте базовый класс Shape с чистой виртуальной функцией area(). Реализуйте два производных класса Circle и Rectangle, которые наследуются от Shape и переопределяют метод area().
Варианты ответа:
a) class Shape { public: virtual double area() const = 0; };
b) class Shape { public: virtual double area() const { return 0; } };
c) class Shape { public: virtual double area() const = 0; };
d) class Shape { public: virtual double area() const { return 0; } };
Правильный ответ:
a) class Shape { public: virtual double area() const = 0; };

5. Что такое статический член класса в C++?
Варианты ответа:
a) Член класса, который принадлежит классу, а не объекту.
b) Член класса, который принадлежит объекту, а не классу.
c) Член класса, который может быть изменен только в базовом классе.
d) Член класса, который может быть изменен только в производном классе.
Правильное решение:
a) Член класса, который принадлежит классу, а не объекту.

6. Что такое константный метод класса в C++?
Варианты ответа:
a) Метод, который не изменяет состояние объекта.
b) Метод, который изменяет состояние объекта.
c) Метод, который может быть вызван только в базовом классе.
d) Метод, который может быть вызван только в производном классе.
Правильное решение:
a) Метод, который не изменяет состояние объекта.

7. Что такое перегрузка операторов в C++?
Варианты ответа:
a) Возможность определения нового поведения для стандартных операторов.
b) Возможность использования операторов только для стандартных типов данных.
c) Возможность использования операторов только для числовых типов данных.
d) Возможность использования операторов только для строковых типов данных.
Правильное решение:
a) Возможность определения нового поведения для стандартных операторов.

8. Что такое шаблон класса в C++?
Варианты ответа:
a) Обобщенное определение класса, которое может работать с разными типами данных.
b) Определение класса, которое работает только с одним типом данных.
c) Определение класса, которое работает только с числовыми типами данных.
d) Определение класса, которое работает только со строковыми типами данных.
Правильное решение:

a) Обобщенное определение класса, которое может работать с разными типами данных.

9. Зачем нужен виртуальный деструктор в C++?
Варианты ответа:
a) Для корректного удаления объектов производного класса через указатель на базовый класс.
b) Для ускорения работы программы.
c) Для уменьшения объема памяти, занимаемой объектом.
d) Для увеличения скорости выполнения деструктора.
Правильное решение:
a) Для корректного удаления объектов производного класса через указатель на базовый класс.

10. Что произойдет при выполнении следующего кода?

```
class Base {
public:
    virtual void show() {
        std::cout << "Base\n";
    }
};

class Derived : public Base {
public:
    void show() override {
        std::cout << "Derived\n";
    }
};

int main() {
    Base* obj = new Derived();
    obj->show();
    delete obj;
    return 0;
}
```

Варианты ответа:
a) Выведется: Base
b)Выведется: Derived
c) Компиляционная ошибка
d) Ошибка времени выполнения
Правильное решение: b)Выведется: Derived

11. Какая часть кода демонстрирует концепцию инкапсуляции?

```
class Account {
private:
    int balance;

public:
    void deposit(int amount) {
        balance += amount;
    }
```

```
    int getBalance() {
        return balance;
    }
};
```

Варианты ответа:

a) Закрытые члены класса
b) Открытые методы
c) Открытые переменные
d) Использование ключевого слова class

Правильное решение: a) Закрытые члены класса

12. Какой модификатор доступа по умолчанию используется для членов этого класса?

```
class Example {
    int x;
public:
    int y;
};
```

a) public
b) protected
c) private
d) Зависит от компилятора

Правильное решение: c) private

13. Напишите, что будет выведено на экран в результате выполнения программы:

```
class A {
public:
    A() {
        std::cout << "Constructor A\n";
    }
};

class B : public A {
public:
    B() {
        std::cout << "Constructor B\n";
    }
};

int main() {
    B b;
    return 0;
}
```

Правильное решение:
Constructor A
Constructor B

14. Какой тип наследования используется в следующем коде?

```
class Parent {
public:
```

```
    void display() {
        std::cout << "Parent\n";
    }
};

class Child : public Parent {};
```

a) Закрытое
b) Защищенное
c) Открытое
d) Виртуальное

Правильное решение: c) Открытое

15. Напишите, какой оператор необходимо перегрузить для работы следующего кода?

```
class Point {
public:
    int x, y;
    Point(int a, int b) : x(a), y(b) {}
};

int main() {
    Point p1(1, 2), p2(3, 4);
    Point p3 = p1 + p2;
    return 0;
}
```

Правильное решение: +

16. Какой принцип ООП демонстрируется в данном коде?

```
class Shape {
public:
    virtual void draw() = 0;
};

class Circle : public Shape {
public:
    void draw() override {
        std::cout << "Circle\n";
    }
};
```

Варианты ответа:
a) Инкапсуляция
b) Наследование
c) Полиморфизм
d) Абстракция

Правильное решение: d) Абстракция

17. Что будет выведено на экран?

```
class A {
public:
    virtual ~A() {
        std::cout << "Destructor A\n";
```

```
    }
};

class B : public A {
public:
    ~B() {
        std::cout << "Destructor B\n";
    }
};

int main() {
    A* obj = new B();
    delete obj;
    return 0;
}
```

Варианты ответа:

a) Destructor A
Destructor B
b) Destructor B
Destructor A
c) Destructor B
d) Ничего не будет выведено

Правильное решение: b) Destructor B

18. Что произойдет, если вызвать метод show() у объекта Derived?

```
class Base {
public:
    virtual void show() = 0;
};

class Derived : public Base {};
```

Варианты ответа:

1. Выведется Base
2. Выведется Derived
3. Компиляционная ошибка
4. Ошибка времени выполнения

Правильное решение: 3

19. Какой из вариантов позволяет создать объект Singleton?

```
class Singleton {
private:
    static Singleton* instance;
    Singleton() {}
public:
    static Singleton* getInstance() {
        if (!instance) {
            instance = new Singleton();
        }
        return instance;
    }
```

};

a) Singleton obj;
b) Singleton* obj = Singleton::getInstance();
c) Singleton obj = Singleton::getInstance();
d) Singleton* obj = new Singleton();

Ответ: b) Singleton* obj = Singleton::getInstance();

20. Какой из приведенных методов является конструктором копирования?

```
class MyClass {
public:
   MyClass(const MyClass& obj);
};
```

Варианты ответа:

a) MyClass(const MyClass& obj);
b) MyClass();
c) void MyClass(MyClass obj);
d) MyClass& operator=(const MyClass& obj);

Правильное решение: a) MyClass(const MyClass& obj);

Задания открытого типа:

1. Что такое валидация модели?
2. Какие дополнительные возможности есть ООП в C# по сравнению с другими языками программирования?
3. Что такое асинхронное программирование

№	Вопрос	Ответ
1	Что такое валидация модели?	Валидация модели — это процесс проверки корректности данных, передаваемых в модель, чтобы убедиться, что они соответствуют заданным правилам и требованиям, таким как тип, диапазон значений или формат.
2	Какие дополнительные возможности есть ООП в C# по сравнению с другими языками программирования?	Вот еще более краткий обзор возможностей ООП в C#: 1. Свойства: Инкапсуляция полей с логикой. 2. Автоматические свойства: Упрощенное создание свойств. 3. Инициализаторы объектов: Легкая инициализация экземпляров. 4. Методы расширения: Добавление методов к существующим типам. 5. Интерфейсы с реализацией по умолчанию: Новые методы без нарушения совместимости. 6. Абстрактные классы и интерфейсы: Гибкость и полиморфизм.

		7. Пространства имен: Организация кода и избежание конфликтов. 8. Статическая типизация: Выявление ошибок на этапе компиляции. 9. События и делегаты: Реализация паттернов проектирования. 10. LINQ: Удобная работа с данными. 11. Асинхронное программирование: Простота работы с асинхронностью. 12. Обработка исключений: Эффективное управление ошибками. Эти функции делают C# мощным и удобным для разработки.
3	Что такое асинхронное программирование.	Асинхронное программирование — это подход к написанию кода, позволяющий выполнять задачи, не блокируя основной поток, чтобы улучшить производительность и отзывчивость приложения, особенно при работе с вводом/выводом или длительными операциями.