

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»

Трехгорный технологический институт –

филиал федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский ядерный университет «МИФИ»

(ТТИ НИЯУ МИФИ)

УТВЕРЖДАЮ

Директор ТТИ НИЯУ МИФИ

_____ Т.И. Улитина

«26» _____ июня 2024 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

«Объектно-ориентированное программирование»

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Профиль: Вычислительные машины, комплексы, системы и сети

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Трехгорный
2024

1. ЦЕЛИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Создание программного продукта – весьма трудоемкая задача, особенно в настоящее время, когда объем программного обеспечения превышает сотни тысяч операторов. Будущий специалист в области разработки программного обеспечения должен иметь представление о современных методах и средствах, используемых для разработки программных систем. Одним из методов разработки сложного программного обеспечения является объектно-ориентированный подход, основанный на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного типа (класса), а классы образуют иерархию с наследованием свойств. В рамках дисциплины «Объектно-ориентированное программирование», относящейся к профессиональному циклу дисциплин, студенты овладевают требуемыми навыками и умениями по проектированию и разработке программного обеспечения с использованием объектно-ориентированного подхода.

1.1. Цели дисциплины

Цели дисциплины «Объектно-ориентированное программирование» – изучение современного подхода к проектированию и разработке программного обеспечения на основе объектно-ориентированной технологии, средств разработки программ на языке программирования C++.

1.2. Задачи дисциплины

Задачей дисциплины «Объектно-ориентированное программирование» является обеспечение подготовки студентов по использованию объектно-ориентированного подхода для разработки программного обеспечения как в процессе обучения в вузе, так и в последующей профессиональной деятельности.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина «Объектно-ориентированное программирование» относится к циклу базовых дисциплин, базируется на знаниях, получаемых студентами из курса «Программирование».

Дисциплина изучается в 3,4 семестрах.

3. КОМПЕТЕНЦИИ СТУДЕНТА, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ / ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ ОБРАЗОВАНИЯ И КОМПЕТЕНЦИИ СТУДЕНТА ПО ЗАВЕРШЕНИИ ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1 Перечень компетенций

Изучение дисциплины «Объектно-ориентированное программирование» направлено на формирование у студентов следующих компетенций:

- способен разрабатывать алгоритмы и программы, пригодные для практического применения (ОПК-8);
- способен осваивать методики использования программных средств для решения практических задач (ОПК-9);
- способен осуществлять организацию и управление проектами в области информатики и вычислительной техники в соответствии с действующими правовыми нормами и требованиями заказчика (ПК-4);
- способен разрабатывать и выполнять отладку программного кода (ПК-8.2).

3.2 Перечень результатов образования, формируемых дисциплиной, с указанием уровня их освоения

В результате освоения дисциплины обучающийся должен:

знать:

- основы объектно-ориентированного подхода к программированию;

- инструментарий для разработки программного обеспечения с применением объектно-ориентированного подхода.

уметь:

- работать с современными системами программирования, включая объектно-ориентированные;
- обеспечивать качество и безопасность создаваемого ПО.

владеть:

- языками процедурного и объектно-ориентированного программирования, навыками разработки и отладки программ не менее чем на одном из алгоритмических процедурных языков программирования высокого уровня;
- навыками организации и проведения экспертного тестирования программ на этапе опытной эксплуатации

3.3 Воспитательная работа

Направление/ цели	Создание условий, обеспечивающих	Использование воспитательного потенциала учебных дисциплин
Профессиональный модуль		
Профессиональное воспитание	- формирование чувства личной ответственности за научно-технологическое развитие России, за результаты исследований и их последствия (В17)	1.Использование воспитательного потенциала дисциплин профессионального модуля для формирования чувства личной ответственности за достижение лидерства России в ведущих научно-технических секторах и фундаментальных исследованиях, обеспечивающих ее экономическое развитие и внешнюю безопасность, посредством контекстного обучения, обсуждения социальной и практической значимости результатов научных исследований и технологических разработок. 2.Использование воспитательного потенциала дисциплин профессионального модуля для формирования социальной ответственности ученого за результаты исследований и их последствия, развития исследовательских качеств посредством выполнения учебно-исследовательских заданий, ориентированных на изучение и проверку научных фактов, критический анализ публикаций в профессиональной области, вовлечения в реальные

		междисциплинарные научно-исследовательские проекты.
	- формирование ответственности за профессиональный выбор, профессиональное развитие и профессиональные решения (B18)	Использование воспитательного потенциала дисциплин профессионального модуля для формирования у студентов ответственности за свое профессиональное развитие посредством выбора студентами индивидуальных образовательных траекторий, организации системы общения между всеми участниками образовательного процесса, в том числе с использованием новых информационных технологий.
	- формирование научного мировоззрения, культуры поиска нестандартных научно-технических/практических решений, критического отношения к исследованиям лженаучного толка (B19)	1.Использование воспитательного потенциала дисциплин/практик "Основы научных исследований", «"Учебная практика (научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)" для: - формирования понимания основных принципов и способов научного познания мира, развития исследовательских качеств студентов посредством их вовлечения в исследовательские проекты по областям научных исследований. 2.Использование воспитательного потенциала дисциплин/практик "Введение в специальность", "Основы научных исследований", "Учебная практика (научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)" для: - формирования способности отделять настоящие научные исследования от лженаучных посредством проведения со студентами занятий и регулярных бесед; - формирования критического мышления, умения рассматривать различные исследования с экспертной позиции посредством обсуждения со студентами современных исследований, исторических предпосылок появления тех или иных открытий и теорий.

	- формирование навыков коммуникации, командной работы и лидерства (B20); - формирование способности и стремления следовать в профессии нормам поведения, обеспечивающим нравственный характер трудовой деятельности и неслужебного поведения (B21); - формирование творческого инженерного/профессионального мышления, навыков организации коллективной проектной деятельности (B22)	1.Использование воспитательного потенциала дисциплин профессионального модуля для развития навыков коммуникации, командной работы и лидерства, творческого инженерного мышления, стремления следовать в профессиональной деятельности нормам поведения, обеспечивающим нравственный характер трудовой деятельности и неслужебного поведения, ответственности за принятые решения через подготовку групповых курсовых работ и практических заданий, решение кейсов, прохождение практик и подготовку ВКР. 2.Использование воспитательного потенциала дисциплин профессионального модуля для: - формирования производственного коллективизма в ходе совместного решения как модельных, так и практических задач, а также путем подкрепление рационально-технологических навыков взаимодействия в проектной деятельности эмоциональным эффектом успешного взаимодействия, ощущением роста общей эффективности при распределении проектных задач в соответствии с сильными компетентностными и эмоциональными свойствами членов проектной группы.
	- формирование культуры информационной безопасности (B23)	Использование воспитательного потенциала дисциплин профессионального модуля для формирования базовых навыков информационной безопасности через изучение последствий халатного отношения к работе с информационными системами, базами данных (включая персональные данные), приемах и методах злоумышленников, потенциальном уроне пользователям.
	УГНС 09.00.00 «Информатика и вычислительная техника»: - формирование навыков цифровой гигиены (B24); - формирование ответственности за обеспечение кибербезопасности (B25); - формирование профессиональной	1. Использование воспитательного потенциала дисциплин "Информатика", "Программирование", "Объектно-ориентированное программирование" для формирования культуры написания и оформления программ, а также привития навыков командной работы за счет использования систем управления проектами и контроля версий. 2.Использование воспитательного потенциала профильных дисциплин для формирования навыков цифровой гигиены, а также системности и гибкости мышления, посредством изучения методологических и

	ответственности, этики и культуры инженера-разработчика информационно-управляющих систем различного назначения, удовлетворяющих современным требованиям к обеспечению безопасности и защиты информации (B26)	технологических основ обеспечения информационной безопасности и кибербезопасности при выполнении и защите результатов учебных заданий и лабораторных работ по криптографическим методам защиты информации в компьютерных системах и сетях. 3. Использование воспитательного потенциала дисциплин профессионального модуля и всех видов практик для формирования приверженности к профессиональным ценностям, ответственности, этике и культуре инженера-разработчика информационно-управляющих систем различного назначения посредством контекстного обучения, осознанного выбора тематики проектов, выполнения индивидуальных и совместных проектов при работе в команде, с последующей публичной презентацией результатов.
--	--	---

4. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 7 зачетных единиц.

№ п/п	Раздел учебной дисциплины	Недели	Виды учебной деятельности, включая самостоятельную работу студентов и трудоемкость (в часах)				Текущий контроль успеваемости и (неделя, форма)	Аттестаци я раздела (неделя, форма)	Макс. балл за раздел *
			Лекции	Лаб. работы	Прак. работы	Самост. работа			
Семестр 3									
1	Раздел 1	1-4	9	4	4	10	T1	КТ1	10
2	Раздел 2	5-8	9	4	4	10	T2	КТ2	15
3	Раздел 3	9-12	9	4	4	10	T3	КТ3	15
4	Раздел 4	13-18	9	6	6	15	T4	КТ4	10
Итого			36	18	18	45			50
Экзамен			27						50
Итого за семестр									100
Семестр 4									
1	Раздел 5	1-4	4	4	9	9	T5	КТ5	10
2	Раздел 6	5-8	4	4	9	9	T6	КТ6	15

3	Раздел 7	9-12	4	4	9	9	T7	КТ7	15
4	Раздел 8	13-18	6	6	9	9	T8	КТ8	10
Итого			18	18	36	36			50
Зачет с оценкой									50
Итого за семестр									100

T – Тест, РГР – Расчетно-графическая работа, УО – Устный опрос, КТ – Контрольная точка

4.1 Содержание лекций

Раздел 1 Введение в объектно-ориентированное программирование

Тема 1.1 Введение. Основные положения объектного подхода к разработке программ.

Тема 1.2 Дополнение C к C++. Комментарии, прототипы, область видимости.

Тема 1.3 Перегрузка функций, значения формальных параметров по умолчанию, ссылки и параметры на ссылки

Тема 1.4 Спецификаторы inline, const, операции new и delete, указатель на void

Раздел 2 Понятие класса. Конструкторы и деструкторы.

Тема 2.1 Описания протокола класса. Классы и методы в языке C++. Ключевое слово this.

Тема 2.2 Функции типа inline. Управление доступом к элементам классов. Функции-члены класса.

Тема 2.3 Конструкторы и деструкторы. Свойства конструкторов и деструкторов.

Тема 2.4 Конструкторы по умолчанию, с параметрами.

Тема 2.5 Конструкторы и массивы объектов

Тема 2.6 Конструктор копирования.

Раздел 3 Дружественные функции.

Тема 3.1 Свойства дружественных функций.

Тема 3.2 Статические члены класса.

Тема 3.3 Различия между дружественными функциями и функциями-членами.

Раздел 4 Перегрузка операций. Наследование.

Тема 4.1 Перегрузка операций. Перегрузка унарных и бинарных операций.

Тема 4.2 Перегрузка унарных и бинарных операций.

Тема 4.3 Подкласс, подтип и принцип подстановки. Формы наследования. Наследование в языке C++.

Тема 4.4 Повторное использование кода. Правила доступа для классов и объектов. Отношение «целое-часть».

Раздел 5. Множественное наследование. Шаблоны классов.

Тема 5.1 Виртуальные функции и полиморфизм. Абстрактные классы.

Тема 5.2 Производные классы: множественное наследование. Виртуальные базовые классы.

Тема 5.3 Шаблоны классов. Параметризация класса для представления массива.

Раздел 6. Шаблоны функций

Тема 6.1 Шаблоны функций. Совпадение сигнатуры и перегрузка.

Раздел 7. Классы исключений. Контейнерные классы.

Тема 7.1 Исключения. Обработка исключительных ситуаций. Классы исключений.

Тема 7.2 Контейнерные классы. Последовательный контейнер. Двусторонняя очередь.

Тема 7.3 Последовательный контейнер. Списки. Стеки и очереди.

Раздел 8. Ассоциативный контейнер. Множества.

Тема 8.1 Ассоциативный контейнер.

Тема 8.2 Множества.

4.2 Тематический план практических работ

1. Технология работы в C++
2. Дополнение C++ к C.
3. Протокол класса.
4. Конструкторы и деструкторы.
5. Конструкторы и массивы объектов, конструктор копирования
6. Дружественные функции.
7. Перегрузка функций и операций.
8. Производные классы: одиночное наследование.
9. Иерархия классов.
10. Множественное наследование.
11. Шаблоны функций.
12. Последовательный контейнер. Двусторонняя очередь.
13. Стеки очереди
14. Ассоциативный контейнер. Множества.
15. Множества.

4.3 Тематический план практических работ

1. Область видимости, перегрузка функций, значения формальных параметров по умолчанию, ссылки и параметры на ссылки

2. Спецификаторы inline, const, операции new и delete, указатель на void.
3. Класс
4. Конструкторы по умолчанию.
5. Конструктор копирования.
6. Дружественные функции. Преобразование типов.
7. Использование дружественных функций.
8. Перегрузка унарных и бинарных операций.
9. Производные классы: одиночное наследование.
10. Абстрактные базовые классы.
11. Параметризация класса для представления массива.
12. Классы исключений.
13. Последовательный контейнер. Списки.
14. Ассоциативный контейнер.
15. Множества.

4.4 Самостоятельная работа студентов

1. Подготовка к лабораторным и практическим работам
2. Проработка лекций
3. Подготовка к рубежному контролю (по темам дисциплины, входящим в раздел).

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Выпускник вуза должен не просто обладать определенной суммой знаний, а уметь при помощи этих знаний решать конкретные задачи производства.

Учитывая требования ОС НИЯУ МИФИ по направлению подготовки 09.03.01 «Информатика и вычислительная техника», реализация компетентностного

подхода должна предусматривать широкое использование в учебном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Лекционные занятия проводятся в специализированной аудитории с применением мультимедийного проектора в виде учебной презентации. Учебные материалы предъявляются обучающимся для ознакомления и изучения, основные моменты лекционных занятий конспектируются. Отдельные темы предлагаются для самостоятельного изучения с обязательным составлением конспекта.

Текущий контроль знаний студентов по отдельным разделам и в целом по дисциплине проводится в форме компьютерного или бумажного тестирования.

В таблице 6 представлены интерактивные образовательные технологии, используемые в аудиторных занятиях.

Таблица 6. Интерактивные образовательные технологии

Семестр	Вид занятия (Л, ПР, ЛР, ТК)	Используемые интерактивные образовательные технологии
3	Л	Мультимедийные технологии
	ПР	Мультимедийные технологии
	ЛР	Мультимедийные технологии
4	Л	Мультимедийные технологии
	ПР	Мультимедийные технологии
	ЛР	Мультимедийные технологии
Всего:		

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО- МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Перечень оценочных средств, используемых для текущей аттестации

Код	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
T1	Тест №1	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Тестовые задания по темам
T2	Тест №2	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Тестовые задания по темам
T3	Тест №3	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Тестовые задания по темам
T4	Тест №4	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
T5	Тест №5	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
T6	Тест №6	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
T7	Тест №7	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
T8	Тест №8	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
КТ1	Контрольная точка №1	Средство проверки умений применять полученные знания для решения расчетно-графических задач определенного типа по теме или разделу	Комплект расчетно-графических заданий по вариантам
КТ2	Контрольная точка №2		

КТ3	Контрольная точка №3		
КТ4	Контрольная точка №4		
КТ5	Контрольная точка №5		
КТ6	Контрольная точка №6		
КТ7	Контрольная точка №7		
КТ8	Контрольная точка №8		

Расшифровка компетенций через планируемые результаты обучения

Связь между формируемыми компетенциями и планируемыми результатами обучения представлена в следующей таблице:

Код	Проектируемые результаты освоения дисциплины и индикаторы формирования компетенций			Средства и технологии оценки
	Знать (З)	Уметь (У)	Владеть (В)	
ОПК-8	31, 32,	У1, У2	В1, В2	Семестры 3 и 4: Т1,Т2,Т3,Т4,Т5, Т6, Т7, Т8, КТ1, КТ2,КТ3,КТ4,КТ5,КТ6,КТ7,КТ8
ОПК-9	31, 32	У1, У2	В1, В2	Семестры 3 и 4: Т1,Т2,Т3,Т4,Т5, Т6, Т7, Т8, КТ1, КТ2,КТ3,КТ4,КТ5,КТ6,КТ7,КТ8
ПК-8.2	31, 32	У1, У2	В1, В2	Семестры 3 и 4: Т1,Т2,Т3,Т4,Т5, Т6, Т7, Т8, КТ1, КТ2,КТ3,КТ4,КТ5,КТ6,КТ7,КТ8

Этапы формирования компетенций

Раздел	Темы занятий	Коды компетенций	Знания, умения и навыки	Виды аттестации		
				Текущий контроль – неделя	Аттестация раздела – неделя	Промежуточная аттестация
3 семестр						
Раздел 1	Введение в объектно-ориентированное программирование	ОПК-8, ОПК-9, ПК8.2	31, 32, 33, У1, У2, У3, В1, В2, В3,	T1	КТ1	Зачет с оценкой
Раздел 2	Понятие класса. Конструкторы и деструкторы	ОПК-8, ОПК-9, ПК8.2	31, 32, 33, У1, У2, У3, В1, В2, В3,	T2	КТ2	
Раздел 3	Дружественные функции	ОПК-8, ОПК-9, ПК8.2	31, 32, 33, У1, У2, У3, В1, В2, В3,	T3	КТ3	
Раздел 4	Перегрузка операций. Наследование	ОПК-8, ОПК-9, ПК8.2	31, 32, 33, У1, У2, У3, В1, В2, В3,	T4	КТ4	
4 семестр						
Раздел 5	Множественное наследование. Шаблоны классов.	ОПК-8, ОПК-9, ПК8.2	31, 32, 33, У1, У2, У3, В1, В2, В3,	T5	КТ5	Экзамен
Раздел 6	Шаблоны функций	ОПК-8, ОПК-9, ПК8.2	31, 32, 33, У1, У2, У3, В1, В2, В3,	T6	КТ6	
Раздел 7	Классы исключений. Контейнерные классы.	ОПК-8, ОПК-9, ПК8.2	31, 32, 33, У1, У2, У3, В1, В2, В3,	T7	КТ7	

Раздел 8	Ассоциативный контейнер. Множества.	ОПК-8, ОПК-9, ПК8.2	31, 32, 33, У1, У2, У3, В1, В2, В3,	Т8	КТ8	
----------	-------------------------------------	---------------------	-------------------------------------	----	-----	--

Шкала оценки образовательных достижений

Код	Вид оценочного средства	Критерии	Балл	Макс. балл– мин. балл
Т1 Т2 Т3 Т4 Т5 Т6 Т7 Т8	Тестовое задание 1,2,3,4,5,6,7,8	выставляется студенту, если 90-100% тестовых вопросов выполнено правильно	10	10 – 7
		выставляется студенту, если 80-89% тестовых задач выполнено правильно	8,5	
		выставляется студенту, если 60-79% тестовых задач выполнено правильно	7	
		при ответе студента менее, чем на 60% вопросов тестовое задание не зачитывается и у студента образуется долг, который должен быть закрыт в течение семестра или на зачетной неделе	<7	
КТ1 КТ2 КТ3 КТ4 КТ5 КТ6 КТ7 КТ8	Контрольная точка 1,2,3,4,5,6,7,8	выставляется студенту, если все сделано правильно	5	5 – 3
		выставляется студенту, если решение содержит ошибки	4	
		выставляется студенту, если решения содержат ошибки и было сдано не в срок	3	
		выставляется студенту, во всех остальных случаях	<3	
Э	Экзамен	выставляется студенту при правильно написанном билете и при ответе на все дополнительные вопросы по курсу с незначительными неточностями, которые студент должен устранить в процессе беседы с преподавателем, в рамках которой он демонстрирует углубленное понимание предмета и владение ключевыми знаниями, умениями и навыками, предусмотренными данной дисциплиной	40-50	50 – 30
		выставляется студенту при правильно написанном билете и при ответе на часть дополнительных вопросов по курсу с демонстраций базовых знаний, умений и навыков, предусмотренных данной дисциплиной	35-39	
		выставляется студенту при написанных ответах на вопросы билета (допускается содержание некоторых неточностей) и	30-34	

		демонстрации базовых знаний, умений и навыков по данной дисциплине		
		если студент не написал ответ хотя бы на один из вопросов билета и не может ответить на дополнительные компетентностно–ориентированные вопросы	<30	

Итоговая оценка представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего и промежуточного контроля и выставляется в соответствии с Положением о кредитно-модульной системе в соответствии со следующей шкалой:

Оценка по 5-балльной шкале	Сумма баллов за разделы	Оценка ECTS
5 – «отлично»	90-100	A
4 – «хорошо»	85-89	B
	75-84	C
	70-74	D
3 – «удовлетворительно»	65-69	E
	60-64	
2 – «неудовлетворительно»	Ниже 60	F

Расшифровка уровня знаний, соответствующего полученным баллам, дается в таблице указанной ниже

Оценка по 5-балльной шкале – оценка по ECTS	Сумма баллов за разделы	Требования к знаниям на экзамене
«отлично» – A	90 ÷ 100	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, использует в ответе материал монографической литературы.
«хорошо» – D, C, B	70 ÷ 89	Оценка «хорошо» выставляется студенту, если он твёрдо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
«удовлетворительно» – E, D	60 ÷ 69	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала.
«неудовлетворительно» –	менее 60	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части

<i>F</i>		программного материала, допускает существенные ошибки. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.
----------	--	---

Контроль остаточных знаний

1. В определении класса члены класса с ключевым словом `private` доступны:

- 1) любой функции программы;
- 2) в случае, если вам известен пароль;
- 3) методам этого класса;
- 4) только открытым членам класса.

2. Операция точки (операция доступа к члену класса) объединяет следующие два элемента (слева направо):

- 1) член класса и объект класса;
- 2) объект класса и класс;
- 3) класс и член этого класса;
- 4) объект класса и член этого класса.

3. Конструктор имеет имя:

- 1) совпадающее с объектом класса;
- 2) произвольное;
- 3) совпадающее с именем класса;
- 4) не имеет имени.

4. Конструктор вызывается автоматически:

- 1) в момент создания объекта;
- 2) в момент разрушения объекта;
- 3) в момент вызова любой член-функции (метода) класса;
- 4) не может вызываться автоматически.

5. Методу класса всегда доступны данные:

- 1) объекта, членом которого он является;
- 2) класса, членом которого он является;
- 3) любого объекта класса, членом которого он является;
- 4) класса, объявленного открытым.

6. Определены три объекта класса. Сколько копий полей класса и методов член-функций) класса содержится в памяти?

- 1) три копии полей и три копии методов;
- 2) по одной копии полей и методов;
- 3) одна копия полей и три копии методов;

4) три копии полей и одна копия методов.

7. Перегрузка операций:

- 1) преобразовывает операции (operator) C++ для работы с объектами класса;
- 2) предоставляет операциям C++ больше, чем они могут обработать;
- 3) даёт новое значение существующим в C++ операциям;
- 4) создает новые операции C++.

8. Пусть существует класс C с объектами obj1, obj2 и obj3.

Выражение obj3 = obj1 - obj2 работает правильно. Здесь перегруженная операция должна:

- 1) создавать именованный временный объект;
- 2) возвращать значение;
- 3) возвращать объект;
- 4) использовать объект, вызвавший операцию, как операнд.

9. Когда вы перегружается операция арифметического присваивания, то результат:

- 1) передается объекту справа от операции;
- 2) передается объекту слева от операции;
- 3) не передается;
- 4) помещён в стек.

10. Для преобразования от определенного пользователем класса к основному типу вы можете использовать:

- 1) встроенную операцию преобразования;
- 2) конструктор с одним аргументом;
- 3) перегруженную операцию =;
- 4) операцию преобразования, являющуюся членом класса.

11. Для преобразования от основного типа к типу, определенному пользователем можно использовать:

- 1) встроенную операцию преобразования;
- 2) конструктор с одним аргументом;
- 3) перегруженную операцию =;
- 4) операцию преобразования, являющуюся членом класса.

12. Назначение наследования состоит в том, чтобы:

- 1) создавать более специализированные классы в более общих;
- 2) передавать аргументы объектам классов;
- 3) добавлять возможности к существующим классам без их модификации;
- 4) улучшать сокрытие данных и их инкапсуляцию.

13. Преимущество использования наследования заключается в:

- 1) обеспечении развития класса путем естественного отбора;
- 2) облегчении создания классов;
- 3) избежании переписывания кода;
- 4) предоставлении полезной информации.

14. Оператор разрешения (::) обычно:

- 1) ограничивает видимость переменных для определенных методов;
- 2) обозначает, от какого базового класса создан производный;
- 3) определяет отдельный класс;
- 4) разрешает неопределенности.

15. Предположим, что класс Drev является частным (private) производным класса Base. Мы определяем объект класса Drev, расположенный в функции main(). Через него мы можем получить доступ к:

- 1) членам класса Drev, объявленным как public;
- 2) членам класса Drev, объявленным как protected;
- 3) членам класса Drev, объявленным как private;
- 4) членам класса Base, объявленным как public;
- 5) членам класса Base, объявленным как protected;
- 6) членам класса Base, объявленным как private.

16. Указатель — это:

- 1) адрес переменной;
- 2) обозначение переменной, которая будет доступна следующей;
- 3) переменная для хранения адреса;
- 4) тип данных для адресных переменных.

17. Выражение `int *var;` - это:

- 1) адрес переменной `var`;
- 2) значения переменной, на которую указывает `var`;
- 3) ссылка для хранения адреса;
- 4) тип данных указателя на `int`.

18. Выражение `*test` означает:

- 1) указатель на переменную `test`;
- 2) ссылку на значение переменной `test`;
- 3) разыменованное значение переменной `test`;
- 4) ссылку на значение переменной, на которую указывает `test`;

19. Тип переменной, на которую указывает указатель, должен присутствовать в определении указателя для того,

- 1) чтобы типы данных не перемешались при выполнении арифметических операций;
- 2) чтобы указатель мог быть использован для доступа к членам структуры;
- 3) чтобы не было затронуто ни одно из религиозных убеждений;
- 4) чтобы компилятор мог правильно выполнять арифметические операции.

20. Операция new:

- 1) возвращает указатель на переменную;
- 2) создает переменную с именем new;
- 3) получает динамическую память для использования;
- 4) позволяет узнать, сколько памяти свободно на данный момент.

21. Иерархия классов:

- 1) показывает те же взаимоотношения, что и схема организации;
- 2) описывает взаимоотношение типа «имеет»;
- 3) описывает взаимоотношения типа «является частью»;
- 4) показывает те же взаимоотношения, что и наследственное дерево.

22. Виртуальные функции позволяют:

- 1) создавать массивы типа «указатель на базовый класс», которые могут содержать указатели на производные классы;
- 2) создавать функции, к которым никогда не будет доступа;
- 3) группировать объекты разных классов так, чтобы они все были доступны с помощью одного и того же вызова функции;
- 4) использовать один и тот же вызов функции для выполнения методов объектов, принадлежащих разным классам.

23. Чистая виртуальная функция - это виртуальная функция, которая:

- 1) не имеет тела;
- 2) не возвращает результата;
- 3) используется в базовом классе;
- 4) не имеет аргументов.

24. Абстрактный класс используется, когда:

- 1) не планируется создавать порожденные классы;
- 2) есть несколько связей между двумя порожденными классами;
- 3) с его помощью запрещено создавать какие-либо объекты;
- 4) вы хотите отложить объявление класса.

25. Дружественная функция может быть использована для того, чтобы:

- 1) разные классы могли пользоваться одними аргументами;
- 2) разрешать доступ к классам, исходный код которых недоступен;
- 3) разрешать доступ к несвязанному классу;
- 4) увеличить многосторонность применения перегружаемой операции.

26. Поток C++:

- 1) представляет собой поток функционального управления;
- 2) представляет собой поток данных из одного места в другое;
- 3) ассоциирован с конкретным классом;
- 4) представляет собой файл.

27. Мы можем выводить текст в объект класса ofstream с использованием оператора вставки « потому, что:

- 1) класс ofstream — это поток;
- 2) оператор вставки работает с любыми классами;
- 3) на самом деле вывод осуществляется в cout;
- 4) оператор вставки перегружен в классе ofstream.

28. Шаблоны позволяют удобным способом создавать семейства:

- 1) переменных;
- 2) функций;
- 3) классов;
- 4) программ.

29. Шаблонный класс:

- 1) создается для того, чтобы храниться в разных контейнерах;
- 2) работает с разными типами данных;
- 3) генерирует идентичные объекты;
- 4) генерирует классы с различным числом методов.

30. Реальный код шаблонной функции генерируется при:

- 1) объявлении функции в исходном коде;
- 2) определении функции в исходном коде;
- 3) вызове функции в исходном коде;
- 4) запуске функции во время работы программы.

31. Исключения передаются:

- 1) из обработчика исключений в контролируемый блок;
- 2) из выражения, генерирующего исключительную ситуацию, в контролируемый блок;
- 3) из точки, где возникла ошибка, в обработчик исключений;

- 4) из выражения, в котором возникла ошибка, в контролируемый блок.

32. Дополнительная информация, передающаяся при генерации исключения, может быть помещена в:

- 1) ключевое слово `throw`;
- 2) функцию, вызвавшую ошибку;
- 3) обработчик исключений;
- 4) объект генерируемого исключения.

Таблица правильных ответов:

№ вопроса	№ ответа
1.	3
2.	4
3.	3
4.	1
5.	1
6.	4
7.	1
8.	3
9.	2
10.	4
11.	2
12.	1
13.	2
14.	4
15.	1
16.	4

№ вопроса	№ ответа
17.	4
18.	4
19.	4
20.	3
21.	4
22.	4
23.	1
24.	3
25.	2
26.	3
27.	4
28.	2,3
29.	2
30.	3
31.	2
32.	4

7 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1 Основная литература

1. Хорев, П.Б. Объектно-ориентированное программирование [Текст]: учебное пособие для студентов, обучающихся по направлению "Информатика и вычислительная техника" / П. Б. Хорев. - 4-е изд., стер. - Москва: Академия, 2012. - 447 с.: ил. - (Высшее профессиональное образование) (Бакалавриат). - Библиогр.: с. 444-445. - ISBN 978-5-7695-9265-2 (в пер.)
2. Бабушкина, И.А. Практикум по объектно-ориентированному программированию [Электронный ресурс] : учебное пособие / И.А. Бабушкина, С.М. Окулов. — Электрон. дан. — М. : Бином. Лаборатория знаний, 2012. — 372 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=8781 — Загл. с экрана.
3. Кузнецов, А.А. Технология объектно-ориентированного программирования. Учебное пособие [Электронный ресурс] : учебное

- пособие / А.А. Кузнецов, М.Ю. Михеев. — Электрон. дан. — Пенза : ПензГТУ (Пензенский государственный технологический университет), 2011. — 70 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=62736 — Загл. с экрана.
4. Каширин И.Ю. От С к С++ [Электронный ресурс]: учебное пособие/ Каширин И.Ю., Новичков В.С.— Электрон. текстовые данные.— М.: Горячая линия - Телеком, 2012.— 334 с.— Режим доступа: <http://www.iprbookshop.ru/12022>.— ЭБС «IPRbooks», по паролю
 5. Комлев Н.Ю. Объектно Ориентированное Программирование. Хорошая книга для Хороших Людей [Электронный ресурс]/ Комлев Н.Ю.— Электрон. текстовые данные.— М.: СОЛОН-ПРЕСС, 2014.— 298 с.— Режим доступа: <http://www.iprbookshop.ru/26923>.— ЭБС «IPRbooks», по паролю
 6. Ашарина И.В. Объектно-ориентированное программирование в С++ [Электронный ресурс]: учебное пособие/ Ашарина И.В.— Электрон. текстовые данные.— М.: Горячая линия - Телеком, 2012.— 320 с.— Режим доступа: <http://www.iprbookshop.ru/12008>.— ЭБС «IPRbooks», по паролю

7.2 Дополнительная литература

7.3 Интернет-ресурсы

№	Наименование ресурса	Интернет-ссылка на ресурс
1	Электронный курс А. Л. Фридман «Язык программирования С++»	http://www.intuit.ru/department/pl/cpp/
2	ключевое слово ООП	http://habrahabr.ru
3	biblioclub	http://biblioclub.ru
4	citforum	http://citforum.ru
5	Sql forum	http://www.sql.ru/forum/cpp

8 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для проведения учебных занятий лекционного и семинарского типа, групповые и индивидуальные консультации, текущего контроля, промежуточной аттестации используются учебные аудитории, оснащенные оборудованием и техническими средствами обучения.

Учебные аудитории для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду вуза.

ТТИ НИЯУ МИФИ обеспечен необходимым комплектом лицензионного и свободно распространяемого программного обеспечения.

Сведения о наличии оборудованных учебных кабинетов, объектов для проведения практических занятий представлены на официальном сайте ТТИ НИЯУ МИФИ: <http://tti-mephi.ru/sveden/objects>