Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Новосибирский национальный исследовательский
государственный университет» (Новосибирский государственный университет, НГУ)

Факультет информационных технологий

СОГЛАСОВАНО
Декан ФИТ НГУ
М.М. Лаврентьев
«03» июля 2019 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Основы объектно-ориентированного программирования

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
Направленность (профиль): Программная инженерия и компьютерные науки
Форма обучения: очная
Год обучения: 2, семестр: 3

	Вид деятельности	Семестр
		3
1	Лекции, час.	32
2	Практические занятия, час.	
3	Лабораторные занятия, час.	32
4	Занятий в контактной форме без учета промежуточной аттестации, час, из них	66
5	в электронной форме, час.	
6	из них аудиторных занятий, час.	64
7	из них в активной и интерактивной форме, час.	32
8	консультаций, час.	2
9	Самостоятельная работа, час.	148
10	в том числе на выполнение письменных работ, час	
11	Форма аттестации (экзамен, зачет, дифференцированный зачет), час	Э 2
12	Всего зачетных единиц[1]	6

Новосибирск 2019

[1] С учетом выделенных часов на промежуточную аттестацию

Рабочая программа дисциплины составлена на основании федерального государственного образовательного стандарта высшего образования по направлению подготовки бакалавров 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА.

Федеральный государственный образовательный стандарт (ФГОС) высшего образования - бакалавриат по направлению подготовки 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА введен в действие приказом Минобрнауки от 19.09.2017 № 929.

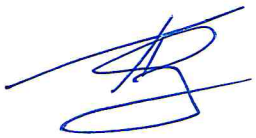
Место дисциплины в структуре учебного плана: Блок 1 Дисциплины (модули), обязательная часть, обязательная дисциплина.

Рабочая программа дисциплины утверждена решением Ученого совета факультета информационных технологий от 02.07.2019, протокол № 75.

Программу разработали:

старший преподаватель кафедры общей информатики ФИТ  В.Ю. Рылов

Заведующий кафедрой общей информатики ФИТ,

доктор физико-математических наук  Д.Е. Пальчунов

Ответственный за образовательную программу:

доцент кафедры систем информатики ФИТ,
кандидат технических наук  А.А. Романенко

Аннотация к рабочей программе дисциплины «Основы объектно-ориентированного программирования»

Дисциплина «Основы объектно-ориентированного программирования» реализуется в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА, направленность (профиль): ПРОГРАММНАЯ ИНЖЕНЕРИЯ И КОМПЬЮТЕРНЫЕ НАУКИ по очной форме обучения на русском языке.

Цели освоения дисциплины:

Дисциплина «Основы объектно-ориентированного программирования» имеет своей целью:

- Изучение основ классической теории объектно-ориентированного программирования, в том числе:
  - Пути эволюции технологий программирования от алгоритмического к ООП
  - Основных принципов объектно-ориентированного построения программных систем (Абстракция, Инкапсуляция, Иерархия, Модульность, Типизация, Параллелизм, Сохраняемость)
  - Понятий классов, объектов, взаимоотношений между ними, а также многоуровневой модели OMG
- Изучение средств объектно-ориентированного и обобщенного программирования языка C++, средств стандартной библиотеки STL
- Выработку практических навыков проектирования и реализации объектно-ориентированных программ на языке программирования C++.

Место в образовательной программе:

Дисциплина «Основы объектно-ориентированного программирования» развивает знания, умения и навыки, сформированные у обучающихся по результатам изучения следующих дисциплин:

- Б1.Б.3 «Информатика»
- Б1.Б.4 «Программирование»
- Б1.Б.7 «Математическая логика и теория алгоритмов»

Предварительными требованиями к студентам являются:

- Знание одного из классических процедурно-ориентированных языков, предпочтительно языка C
- Знания в области алгоритмической декомпозиции, основных структур данных и технологий работы с ним
- Знание основ теории множеств

Дисциплина «Основы объектно-ориентированного программирования» является базовой для освоения следующих дисциплин:

- Б1.В.ОД.3 «Объектно-ориентированное программирование на Java»
- Б1.В.ОД.7 «Объектно-ориентированный анализ и дизайн»
- Б1.Б.24 «Базы данных»
- Б1.В.ДВ.3.3 «Инженерная и компьютерная графика»

Дисциплина «Основы объектно-ориентированного программирования» реализуется в 3 семестре в рамках обязательной части дисциплин (модулей) Блока 1 и является обязательной дисциплиной.

Содержание дисциплины охватывает круг вопросов, связанных с теорией объектно-ориентированного программирования и особенностями ее поддержки и реализации основных принципов в языке программирования C++ и его стандартной библиотеке

По окончании курса студенты получают следующие знания и навыки:

- Знание основ технологии объектно-ориентированной декомпозиции программных систем, базовых шаблонов проектирования (Наблюдатель, Итератор, Одиночка, Фабрика, Заместитель), отношений между классами и основ UML (диаграммы классов и последовательностей).
- Знание особенностей построения объектно-ориентированных программных систем на C++.
- Основные инструментальные средства языка C++ и стандартной библиотеки
- Знания и навыки использования системы и библиотеки автоматизированного тестирования Google Test Framework для C++
- Навыки использования среды Microsoft Visual Studio C++

Дисциплина «Основы объектно-ориентированного программирования» направлена на формирование компетенций:

Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности (ОПК-1), в части следующих индикаторов достижения компетенции:

ОПК-1.1 Знать: основы математики, физики, вычислительной техники и программирования

ОПК-1.2 Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования

ОПК-1.3 Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности

Способен разрабатывать алгоритмы и программы, пригодные для практического применения (ОПК-8), в части следующих индикаторов достижения компетенции:

ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения

ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули

ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы

Перечень основных разделов дисциплины:

1. Эволюция методологий программирования. Основные принципы объектно-ориентированного программирования
2. Объектно-ориентированная модель
3. Классы, метамодель и основы UML
4. Основные алгоритмические отличия C++ от C
5. Средства объектного программирования языка C++
6. Средства объектно-ориентированного программирования C++
7. Обобщенное программирование

8. Стандартная библиотека C++

В процессе изучения дисциплины студенты прослушивают лекции, презентации которых доступные в электронном виде, дополнительно самостоятельно изучают темы с использванием дополнительной литературы и рекомендуемых ресурсов сети Интернет.

В курсе изучается современный стандарт языка программирования C++ ISO/IEC 14882:2017.

Лекционные занятия на курсе проводятся с использованием мультимедийного проектора и в сопровождении с презентациями в формате Power Point.

Дополнительно на лекциях проводятся демонстрации работы основных средств языков/платформ с использованием среды разработки и отладчика.

Лабораторные занятия проходят в терминальных классах, оснащенных персональными компьютерами с установленными средами разработки для C++. Допускается использованием студентами собственных персональных компьютеров (ноутбуков).

Во время лабораторных занятий студенты совместно с преподавателем разбирают вопросы по теме курса и занятий, прорабатывают методику решения практических заданий (сформулированных в форме задач на разработку программ на языке C++), решают лабораторные задания путем разработки программ в процессе самоподготовки. По решению заданий студенты здают и защищают разработанные программы преподавателю.

Дополнительно преподаватели по желанию могут осуществлять прием и проверку заданий по электронной почте.

В процессе самостоятельной подготовки студенты готовятся к экзамену и имеют возможность задавать вопросы во время предэкзаменационной консультации.

Общий объем дисциплины –6 зачетных единиц (216 часов).

Правила аттестации по дисциплине

Текущий контроль по дисциплине «Основы объектно-ориентированного программирования» осуществляется во время проведения лабораторных занятий в следующей форме:

- За решение и сдачу практических задач студенту начисляются баллы, определяющие успеваемость на лабораторных занятиях в течение семестра.
- В течение семестра проводится тестирование в форме коллоквиума (теста с вопросами, подразумевающими открытую форму ответа)

Сдача лабораторной работы (задачи) подразумевает демонстрацию сборки разработанной программы из исходных кодов на языке программирования C++ и демонстрации ее работы в соответствии с требованиями лабораторного задания, прохождение автоматических тестов, ответы на вопросы по коду с целью подтверждения авторства.

Результаты выполнения лабораторных работ и ответов на вопросы коллоквиума формируют портфолио.

Промежуточная аттестация по дисциплине «Основы объектно-ориентированного программирования» проводится по завершению каждого периода ее освоения (семестра) в форме экзамена.

Количество набранных баллов за сдачу лабораторных работ и коллоквиума (портфолио) является важным критерием при выставлении оценки на экзамене и является одним из условий прохождения промежуточной аттестации.

Экзамен проходит в устной форме по вопросам экзаменационного билета. В процессе сдачи экзамена студенту могут задаваться дополнительные задания по теме вопросов билета в форме написания фрагмента кода демонстрирующего определенный механизм языка программирования или технику объектно-ориентированного программирования на C++.

Результаты промежуточной аттестации по дисциплине оцениваются по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Оценка «отлично» соответствует продвинутому уровню сформированности компетенции.

Оценка «хорошо» соответствует базовому уровню сформированности компетенции.

Оценка «удовлетворительно» соответствует пороговому уровню сформированности компетенции.

Учебно-методическое обеспечение дисциплины

Учебно-методический комплекс по дисциплине «Основы объектно-ориентированного программирования» размещен на сайте http://sites.google.com/site/nguoop

1. Внешние требования к дисциплине

Таблица 1.1

Компетенция ОПК-1: Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности*, в части следующих индикаторов достижения компетенции:*
ОПК-1.1 Знать: основы математики, физики, вычислительной техники и программирования
ОПК-1.2 Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и обще-инженерных знаний, методов математического анализа и моделирования
ОПК-1.3 Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности
Компетенция ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения*, в части следующих индикаторов достижения компетенции:*
ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения
ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули
ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы

2. Требования к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий		
	Лекции	Лабораторные	Самостоятельная работа
ОПК-1: Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности, в части следующих результатов обучения:			
ОПК-1.1 Знать: основы математики, физики, вычислительной техники и программирования			
1. Знать принципы и основные элементы объектной модели разработки программного обеспечения (абстракция, инкапсуляция, иерархия, модульность, типизация, параллелизм, сохраняемость)	+	+	+
2. Знать нотацию языка моделирования UML в части диаграмм классов, диаграмм последовательности	+	+	+
ОПК-1.2 Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования			
3. Уметь разрабатывать и визуализировать модель классов системы на языке UML		+	+
ОПК-1.3 Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности			
4. Иметь представление об основных современных средах разработки для C++ для разных платформ (Windows, Linux, Mac OS X) – Microsoft Visual Studio, JetBrains		+	+

CLion, Qt Creator и др. Владеть одной из сред разработки по выбору (Microsoft Visual Studio либо JetBrains CLion)			
ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения, в части следующих результатов обучения:			
ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения			
5. Знать основные функции системы автоматической сборки CMake		+	+
6. Знать основные средства поддержки процедурного стиля программирования (типы данных, переменные, структура программы, функции, структурные типы) языка программирования C++	+	+	+
7. Знать средства языка C++ поддерживающие основные принципы объектно-ориентированного подхода	+	+	+
8. Знать средства обобщенного программирования на C++ (шаблоны)	+	+	+
9. Знать основные средства и принципы организации стандартной библиотеки C++	+	+	+
10. Знать технологию автоматического модульного тестирования программ на C++ с использованием библиотеки Google Test		+	+
ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули			
11. Уметь осуществлять объектно-ориентированную декомпозицию программ на языке C++ с разделением на модули с последующей реализацией		+	+
12. Уметь разрабатывать и реализовывать автоматизированные тесты с целью верификации корректности реализованных программ с использованием библиотеки Google Test		+	+
ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы			
13. Владеть основными средствами языка программирования C++, решать практические задачи с его использованием		+	+
14. Владеть системой тестирования Google Test, навыками отладки и тестирования программных систем с ее использованием		+	+

3. Содержание и структура учебной дисциплины

Таблица 3.1

№ лекции	Темы лекций	Активные формы, час	Часы	Ссылки на результаты обучения
1	1. Основные принципы объектно-ориентированного программирования 1.1. Эволюция методологий программирования 1.1.1. Начало начал, или первое поколение языков программирования. 1.1.2. Развитие алгоритмических абстракций, или второе поколение языков программирования. 1.1.3. Модуль как единица построения программных систем, третье поколение языков программирования. 1.1.4. Зарождение объектной модели, четвертое поколение языков программирования. 1.1.5. Объектные языки программирования, объектно-ориентированные языки программирования, объектно-ориентированный анализ, дизайн и проектирование. 1.1.6. Парадигмы программирования. 1.2. Составные части объектного подхода 1.2.1. Абстрагирование 1.2.2. Инкапсуляция 1.2.3. Модульность 1.2.4. Иерархия 1.2.5. Типизация 1.2.6. Параллелизм 1.2.7. Сохраняемость	1	2	1
2	2. Объектно-ориентированная модель 2.1. Понятие объекта 2.2. Свойства, присущие объектам 2.2.1. Состояние 2.2.2. Поведение 2.2.3. Идентичность 2.3. Отношения между объектами 2.3.1. Типы отношений 2.3.2. Связь (ассоциация) 2.3.3. Агрегация	1	2	1
3	3. Классы 3.1. Природа классов. 3.2. UML – унифицированный язык моделирования. Четырехуровневая метамодель MOF 3.3. Отношения между классами. 3.3.1. Типы отношений 3.3.2. Ассоциация 3.3.3. Агрегация 3.3.4. Использование	1	2	1, 2

	3.3.5. Наследование 3.3.6. Инстанцирование 3.4. Отношения между классами и объектами			
4	4. Основные алгоритмические отличия C++ от C 4.1. Использование ссылок. Передача аргументов функции по ссылке. 4.2. Использование констант. 4.3. Логические тип и перечисления. 4.4. Операторы управления динамической памятью, инициализация массивов. 4.5. Структура программы, раздельная компиляция и особенности использования статической памяти. 4.6. Пространства имен и исключения (краткий обзор) 4.7. Библиотека ввода вывода (краткий обзор iostream) 4.8. Функциональный полиморфизм. 4.9. Лямбды, замыкания и указатели на функции	1	2	6
5 - 8	5. Средства объектного программирования языка C++ 5.1. Представление объектов и классов. 5.1.1. Реализация поведения объектов на примере добавления функций—членов в структуры. Структура как вырожденный класс. 5.1.2. Структура объявления класса. 5.1.3. Доступ к членам класса. 5.1.4. Поля данных класса как механизм реализации состояния объекта. 5.1.5. Функции члены класса как механизм реализации поведения объекта. 5.1.6. Спецификаторы доступа для обеспечения инкапсуляции. 5.1.7. Средства управления жизнью объекта. Конструкторы и деструкторы. 5.1.8. Конструирование и уничтожение объектов и массивов объектов. 5.1.9. Особенности использования конструктора копии, конструктора по умолчанию, оператора присваивания. 5.1.10. Описание селекторов и модификаторов. 5.1.11. Перегрузка операторов C++ как реализация поведения с предопределенным смыслом. 5.1.12. Дружественность как механизм нарушения инкапсуляции. Достоинства и недостатки механизма дружественности. 5.1.13. Статические поля и методы классов. Инициализация статических полей.	4	8	7

	5.2. Реализация отношений между объектами и классами 5.2.1. Ассоциация и агрегация объектов и классов. Зависимость по времени жизни. 5.2.2. Использование и зависимость от интерфейсов. 5.2.3. Объекты при передаче параметров и возврате из методов. 5.2.4. Варианты реализации отношения клиент-сервер. 5.2.5. Внутренние классы. 5.3. Средства модульности 5.3.1. Разбиение программы на модули, типы связывания 5.3.2. Объектная структура модуля, утилиты для анализа модулей, архивные библиотеки 5.3.3. Компоновка программ и динамических библиотек, .ELF и .COFF, особенности Windows и Linux 5.3.4. Поддержка многомодульных проектов в средах разработки 5.4.5. Системы автоматической сборки, Make, Autoconf, CMake			
9 - 11	6. Средства объектно-ориентированного программирования C++ 6.1. Наследование как средство организации иерархий классов. Принцип замещения Лисковой. 6.2. Одиночное наследование. 6.2.1. Понятие производного класса. 6.2.2. Управление доступом в производных классах. 6.2.3. Конструкторы и деструкторы, совмещение имен методов при наследовании, иерархии. 6.2.4. Абстрактные классы и виртуальные функции. 6.2.5. Виртуальный полиморфизм. 6.2.6. Информация о типе на этапе выполнения. RTTI. 6.3. Множественное наследование 6.3.1. Проблема множественного наследования. Виртуальное наследование как средство разрешения коллизий. 6.3.2. Порядок вызовов конструкторов и деструкторов при множественном наследовании. 6.3.3. Чистые виртуальные классы, понятие интерфейса. 6.3.4. Принципы дизайна иерархий классов. OCP, DIP, ISP. 6.4. Пространства имен. 6.4.1. Пространства имен как средство	3	6	7

	реализации модульности. 6.4.2. Поиск имен и разрешение конфликтов. 6.4.3. Объединение пространств имен. 6.4.4. Принципы дизайна пакетов. 6.5. Обработка исключений. 6.5.1. Обработка ошибок. 6.5.2. Группировка и перехват исключений. 6.5.3. Управление ресурсами. 6.5.4. Исключения и эффективность. 6.5.5. Альтернативные методы обработки ошибок. 6.5.6. Стандартные исключения.			
12 - 13	7. Обобщенное программирование. 7.1. Шаблоны классов. 7.1.1. Определение шаблона. 7.1.2. Инстанцирование. 7.1.3. Параметры шаблонов и проверка типов. 7.2. Шаблоны функций. 7.3. Специализация. 7.4. Наследование и шаблоны.	2	4	8
14 - 16	8. Стандартная библиотека C++ 8.1. Библиотека стандартных шаблонов 8.1.1. Общие сведения (понятия контейнеров, итераторов и объектов-функций) 8.1.2. Контейнеры (виды контейнеров, последовательные и ассоциативные контейнеры, адаптеры) 8.1.3. Итераторы (итератор как обобщение указателя, классы итераторов) 8.1.4. Алгоритмы (примеры алгоритмов с использованием итераторов: алгоритмы сортировки, алгоритмы, не изменяющие содержание контейнера, алгоритмы, изменяющие содержание контейнера) 8.2. Библиотека ввода-вывода 8.2.1. Потоки вывода. Вывод типов определяемых пользователем. 8.2.2. Потоки ввода. Ввод типов определяемых пользователем. 8.2.3. Форматирование в потоках ввода-вывода. 8.2.4. Буферизация.	3	6	9
	Итого	**16**	**32**	

Таблица 3.2

Темы лабораторных занятий	Актив-ные формы, час	Часы	Ссылки на результаты обучения	Учебная деятельность
Тема 1. Раздельная компиляция и пространства имен	3	6	4, 6, 11,13	Изучение методических указаний, обсуждение с преподавателем, реализация и

				сдача программы задания №1
Тема 2. Перегрузка функций, указатели на функции, перечисления	1	2	4, 6, 9, 11, 13	Изучение методических указаний, обсуждение с преподавателем. Реализация и сдача программы задания № 2
Тема 3. Классы в языке C++, библиотека Google Test, Система сборки CMake	4	8	4, 5, 6, 9, 10, 11, 13	Изучение методических указаний, обсуждение с преподавателем. Реализация и сдача программы задания № 3, задания №4
Тема 4. Иерархии классов, наследование	3	6	1, 2, 3, 4, 5, 6, 7, 9, 10, 12, 13, 14	Изучение методических указаний, обсуждение с преподавателем. Реализация и сдача программы задания № 4
Тема 5. Шаблоны, обобщенное программирование, стандартная библиотека	5	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14	Изучение методических указаний, обсуждение с преподавателем. Реализация и сдача программы задания № 5
Итого	**16**	**32**		

4. Самостоятельная работа студентов

Таблица 4.1

№	Виды самостоятельно работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
1	Подготовка к лабораторным занятиям по теме 1, реализация заданий лабораторных работ	4, 6, 11, 13	24	
	Обучающиеся изучают материалы лекций, опубликованные в электронном виде, читают дополнительную литературу, изучают методические указания к заданию, опубликованные на сайте курса, пользуются справочными ресурсами сети Интернет			
2	Подготовка к лабораторным занятиям по теме 2, реализация заданий лабораторных работ	4, 6, 9, 11, 13	10	
	Обучающиеся изучают материалы лекций, опубликованные в электронном виде, читают дополнительную литературу, изучают методические указания к заданию, опубликованные на сайте курса, пользуются справочными ресурсами сети Интернет			
3	Подготовка к лабораторным занятиям по теме 3, реализация заданий лабораторных работ	4, 5, 6, 9, 10, 11, 13	32	
	Обучающиеся изучают материалы лекций, опубликованные в электронном виде, читают дополнительную литературу, изучают методические указания к заданию, опубликованные на сайте курса, пользуются справочными ресурсами сети Интернет			
4	Подготовка к лабораторным занятиям по теме 4, реализация заданий лабораторных работ	1, 2, 3, 4, 5, 6, 7, 9, 10, 12, 13, 14	24	
	Обучающиеся изучают материалы лекций, опубликованные в электронном виде, читают дополнительную литературу, изучают методические указания к заданию, опубликованные на сайте курса, пользуются справочными ресурсами сети Интернет			
5	Подготовка к лабораторным занятиям по	1, 2, 3, 4, 5, 6,	34	

	теме 5, реализация заданий лабораторных работ	7, 8, 9, 10, 12, 13, 14		
	Обучающиеся изучают материалы лекций, опубликованные в электронном виде, читают дополнительную литературу, изучают методические указания к заданию, опубликованные на сайте курса, пользуются справочными ресурсами сети Интернет			
6	Подготовка к экзамену	1, 2, 6, 7, 8, 9	24	2
	Подготовка ко второму этапу промежуточной аттестации по вопросам, представленным в фонде оценочных средств, являющихся приложением к рабочей программе дисциплины.			
	Итого		**148**	**2**

5. Образовательные технологии

Лекционные занятия на курсе проводятся с использованием мультимедийного проектора и в сопровождении с презентациями в формате Power Point, а также с демонстрациями исходного кода и работы программ в среде разработки Microsoft Visual Studio, в том числе с использованием отладчика.

В курсе изучается современный стандарт языка программирования C++ ISO/IEC 14882:2017.

В процессе лекции студентам предлагаются вопросы для коллективного обсуждения и анализа, студенты имеют возможность активно задавать вопросы. Продуктивной является форма анализа в посылке «Как бы сделал я если бы разрабатывал язык и компилятор».

Для особо интересных и сложных вопросов используется коллективное голосование в режиме «кто за вариант А», а теперь «кто за вариант Б» с последующим анализом и объяснением «как оно сделано на самом деле и почему».

Данный подход позволяет дать студентам почуствовать саму философию языка C++ и мотивы, движущие его созданием и развитием как языка «созданного программистом для программистов» и «полезного здесь и сейчас», по словам его создателя Б. Страуструпа.

Подобная интерактивная форма концентрирует внимание слушателей и позволяет лектору лучше чувствовать степень понимания материала студентами с возможностью корректировки стиля и глубины изложения.

Лабораторные занятия проходят в терминальных классах, оснащенных персональными компьютерами с установленными средами разработки для C++. Допускается использование студентами собственных персональных компьютеров (ноутбуков).

Во время лабораторных занятий студенты совместно с преподавателем разбирают вопросы по теме курса и занятия, прорабатывают методику решения практических заданий (сформулированных в форме задач на разработку программ на языке C++), решают лабораторные задания путем разработки программ в процессе самоподготовки. По решению заданий студенты сдают и защищают разработанные программы преподавателю.

Дополнительно преподаватели по желанию могут осуществлять прием и проверку заданий по электронной почте.

В процессе самостоятельной подготовки студенты готовятся к экзамену и имеют возможность задавать вопросы во время предэкзаменационной консультации.

Для организации и контроля самостоятельной работы студентов, а также проведения консультаций применяются информационно-коммуникационные технологии.

Таблица 5.1

Информирование	Группы рассылки по электронной почте формируемые семинаристами для каждой группы.

	Сайт курса по адресу http://sites.google.com/site/nguoop
Консультирование	Электронная почта лектора (v.rylov@g.nsu.ru), электронная почта семинаристов
Контроль	Электронные ведомости учета успеваемости и посещаемости размещаемые на платформе Google docs (http://docs.google.com), репозитории системы контроля версий на платформе bitbucket.org (http://bitbucket.org)
Размещение учебных материалов	Сайт курса по адресу http://sites.google.com/site/nguoop

6. Правила аттестации студентов по учебной дисциплине

По дисциплине «Основы объектно-ориентированного программирования» проводится текущая и промежуточная аттестация (итоговая по дисциплине)

Текущий контроль по дисциплине «Основы объектно-ориентированного программирования» осуществляется во время проведения лабораторных занятий в следующей форме:

- За решение и сдачу практических задач студенту начисляются баллы, определяющие успеваемость на лабораторных занятиях в течение семестра.
- В течение семестра проводится тестирование в форме коллоквиума (теста с вопросами подразумевающими открытую форму ответа)

Сдача лабораторной работы (задачи) подразумевает демонстрацию сборки разработанной программы из исходных кодов на языке программирования C++ и демонстрации ее работы в соответствии с требованиями лабораторного задания, прохождение автоматических тестов, ответы на вопросы по коду с целью подтверждения авторства.

Решенные и успешно сданные лабораторные работы совместно с ответами на вопросы коллоквиума формируют портфолио обучающегося.

С целью контроля прогресса решения лабораторных заданий студенты сохраняют исходный код в процессе работы над практическими заданиями в системе контроля версий на основе технологии git или mercurial на общедоступных репозиториях в сети интернет. Выбор репозитория осуществлвяется на усмотрение студента по согласованию с преподавателем (типичным выбором является общеизвестный репозиторий bitbucket.org со свободным планом обслуживания для маленьких проектов).

Промежуточная аттестация по дисциплине «Основы объектно-ориентированного программирования» проводится по завершению каждого периода ее освоения (семестра) в форме экзамена.

Количество набранных баллов за сдачу лабораторных работ и коллоквиума является важным критерием при выставлении оценки на экзамене и является одним из условий прохождения промежуточной аттестации.

Экзамен проходит в усной форме по вопросам экзаменационного билета. В процессе сдачи экзамена студенту могут задаваться дополнительные задания по теме вопросов билета в форме написания фрагмента кода демонстрирующего определенный механизм

языка программирования или технику объектно-ориентированного программирования на C++.

Результаты промежуточной аттестации по дисциплине оцениваются по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

В таблице 6.1 представлено соответствие форм аттестации заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.1

Коды компетенций ФГОС	Результаты обучения	Формы аттестации	
		Портфолио	Экзамен
ОПК-1	ОПК-1.1 Знать:основы математики, физики, вычислительной техники и программирования	+	+
ОПК-1	ОПК- 1.2 Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования	+	+
ОПК-1	ОПК-1.3 Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности	+	
ОПК-8	ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения	+	+
ОПК-8	ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы , интегрировать программные модули	+	
ОПК-8	ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы	+	

Оценочные средства, а также критерии оценки сформированности компетенций и освоения дисциплины в целом, представлены в Фонде оценочных средств, являющемся приложением 1 к настоящей рабочей программе дисциплины.

7. Литература

1. Страуструп, Б. Язык программирования C++ для профессионалов / Б. Страуструп. — 2-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 670 c. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/73737.html
2. Мухортов В.В., Рылов В.Ю. Объектно-ориентированное программирование, анализ и дизайн. Методическое пособие. ИМ СО РАН, 2002 г. (электронная версия в формате

PDF свободно доступна по адресу https://sites.google.com/site/nguoop/materialy-lekcij---s/OOP%26OOD.PDF?attredirects=0&d=1)

Электронные ресурсы сети Интернет

3. Справочный сайт по языку C++ и стандартной библиотеке (на английском языке), свободный доступ: http://www.cplusplus.com/
4. Справочный сайт по языку C++ и стандартной библиотеке (многоязычная версия), свободный доступ: http://www.cppreference.com/
5. Справочный сайт Microsoft по Visual Studio и C++ (многоязычная версия), свободный доступ: https://docs.microsoft.com/ru-ru/cpp/?view=vs-2017
6. Справочный сайт по системе сборки CMake (на английском языке), свободный доступ: https://gitlab.kitware.com/cmake/community/wikis/home
7. Справочные материалы по системе автоматического модульного тестирования Google test (на английском языке), свободный доступ: https://github.com/google/googletest/tree/master/googletest/docs

8. Учебно-методическое и программное обеспечение дисциплины

8.1. Учебно-методическое обеспечение

Основным учебно-образовательным ресурсом курса является WWW сайт http://sites.google.com/site/nguoop

На данном сайте представлены:

- Правила учета успеваемости
- Посещаемость лекций в текущем учебном году
- Демонстрационные презентации лекций курса в формате Microsoft Power Point
- Демонстрационные примеры программ, представленные на лекциях
- Условия практических заданий и курсовых работ для текущего учебного года
- Список основной и дополнительной литературы
- Список вопросов для самоподготовки к экзамену

8.2. Программное обеспечение

Для обеспечения реализации дисциплины используется стандартный комплект программного обеспечения (ПО), включающий регулярно обновляемое лицензионное ПО Windows.

Перечень специализированного программного обеспечения для изучения дисциплины представлен в таблице 8.1.

Специализированное программное обеспечение

Таблица 8.1

	Наименование ПО	Назначение
1	Microsoft Visual Studio Professional 2019	Среда разработки приложений
2	Jet Brains CLion 2018	Среда разработки приложений

9. Профессиональные базы данных и информационные справочные системы

При изучении дисциплины не используются

10. Материально-техническое обеспечение

Таблица 10.1

	Наименование	Назначение
1	Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления)	Для проведения лекционных занятий
2	Компьютерный класс (с выходом в Internet)	Для организации лабораторных занятий и самостоятельной работы обучающихся

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».

Лист актуализации рабочей программы дисциплины
«Основы объектно-ориентированного программирования»

№	Характеристика внесенных изменений (с указанием пунктов документа)	Дата и № протокола Ученого совета ФИТ	Подпись ответственного
1.	Актуализирована на 2020-2021 уч.год	22.07.2020 №77	

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Новосибирский национальный исследовательский
государственный университет» (Новосибирский государственный университет, НГУ)

Факультет информационных технологий

СОГЛАСОВАНО
Декан ФИТ НГУ
_____ М.М. Лаврентьев
«03» июля 2019 г.

Фонд оценочных средств промежуточной аттестации

по дисциплине **Основы объектно-ориентированного программирования**

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Программная инженерия и компьютерные науки

Квалификация: бакалавр

Форма обучения: очная

Год обучения: 2, семестр 3

Форма аттестации	Семестр
Экзамен	3

Новосибирск 2019

Фонд оценочных средств промежуточной аттестации по дисциплине является **Приложением 1** к рабочей программе дисциплины «Основы объектно-ориентированного программирования», реализуемой в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 Информатика и вычислительная техника, направленность (профиль): Программная инженерия и компьютерные науки

Фонд оценочных средств промежуточной аттестации по дисциплине утвержден решением ученого совета факультета информационных технологий, протокол № 75 от 02.07.2019.

Разработчики:
старший преподаватель кафедры общей информатики ФИТ, В.Ю. Рылов

Заведующий кафедрой общей информатики ФИТ,
доктор физико-математических наук Д.Е. Пальчунов

Ответственный за образовательную программу:
доцент кафедры систем информатики ФИТ,
кандидат технических наук А.А. Романенко

1. Содержание и порядок проведения промежуточной аттестации по дисциплине

1.1. Общая характеристика содержания промежуточной аттестации

Промежуточная аттестация по дисциплине «Основы объектно-ориентированного программирования» проводится по завершению периода освоения образовательной программы (семестра) для оценки сформированности компетенций в части следующих индикаторов достижения компетенции (таблица П1.1).

Таблица П1.1

Код	Компетенции, формируемые в рамках дисциплины	Семестр 3	
		Портфолио	Экзамен
ОПК-1: Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности, в части следующих результатов обучения:			
ОПК-1.1	Знать: основы математики, физики, вычислительной техники и программирования	+	+
ОПК-1.2	Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования	+	+
ОПК-1.3	Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности	+	
ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения, в части следующих результатов обучения:			
ОПК-8.1	Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения	+	+
ОПК-8.2	Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули	+	
ОПК-8.3	Владеть: языком программирования; навыками отладки и тестирования работоспособности программы	+	

Промежуточная аттестация включает 2 этапа. Часть компетенций оценивается портфолио, в которое входят работы, выполненные в рамках дисциплины. Часть компетенций оценивается экзаменом.

Тематика экзаменационных вопросов и заданий экзамена включает следующие темы (разделы) в двух категориях:

Категория 1 – Основы объектного подхода

- Эволюция стилей и поколения языков программирования
- Основные принципы объектно-ориентированного подхода (абстракция, инкапсуляция, модульность, иерархия, типизация, параллелизм, сохраняемость)
- Объект с точки зрения ООП, взаимоотношения между объектами
- Класс с точки зрения ООП, взаимоотношения между классами
- Многоуровневая метамодель

Категория 2 – Средства объектно-ориентированного программирования C++

- Средства поддержки процедурного стиля в языке C++
- Средства поддержки объектно-ориентированного подхода в C++
- Средства поддержки обобщенного программирования в C++
- Основы STL и стандартная библиотека C++

1.2. Порядок проведения промежуточной аттестации по дисциплине

Промежуточная аттестация проводится в форме экзамена и включает 2 этапа: портфолио и экзамен. Необходимым условием для прохождения промежуточной аттестации является оценка «зачтено» по результатам выполненного портфолио. Для оценивания портфолио студенту необходимо сдать все работы, входящие в структуру портфолио.

Оценка портфолио осуществляется по балловой системе.

При защите задания обучающийся должен:

- Изложить необходимый для решения теоретический материал
- Указать методику решения задания, предоставить диаграмму классов в нотации UML
- Предоставить исходный код программы или программ реализующий требования задания
- Продемонстрировать корректную работу программы и прохождения автоматических модульных тестов.

За решенные задания начисляются баллы в соответствии с условиями задач. В случае сдачи задания с недочетами или позже установленного срока начисляются баллы меньше балловой стоимости задачи, но не менее 50% при условии успешного решения.

Коллоквиум состоит из 18-20 вопросов, проводится в виде письменного теста, за каждый правильный ответ на вопрос коллоквиума начисляется от 1 до 2 баллов в зависимости от сложности вопроса.

Для получения оценки зачтено необходимо набрать не менее 65 баллов при типичной стоимости задачи от 10 до 30 баллов. Должно быть сдано не менее 4 заданий.

Экзамен проводится в устной форме. Во время проведения экзамена студенту не разрешается использовать технические средства и литературу. В процессе ответа на вопросы экзаменационного билета студенту могут быть заданы дополнительные вопросы по темам дисциплины

2. Требования к структуре и содержанию фонда оценочных средств промежуточной аттестации по дисциплине

Перечень оценочных средств, применяемых на каждом этапе проведения промежуточной аттестации по дисциплине, представлен в таблице П1.2.

Таблица П1.2

№ п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
Семестр 3			
1	Коллоквиум	Средство контроля усвоения учебного материала дисциплины, организованное как письменный тест	Вопросы по темам/разделам дисциплины
2	Портфолио	Целевая подборка работ студента, рас-	Структура портфо-

		крывающая его индивидуальные образовательные достижения в одной или нескольких учебных дисциплинах.	лио
3	Экзаменационный билет	Комплекс вопросов и/или разноуровневых заданий (списать подходящее из п. про задачи)	Список теоретических вопросов и задач

2.1. Требования к структуре и содержанию оценочных средств аттестации

2.1.1. Вопросы коллоквиума

1) Сформулируйте основную идею объектно-ориентированного программирования. В чем его преимущества перед процедурным стилем?
2) В чем состоит принцип абстракция в ООП? Какие средства языка С++ поддерживают данный принцип?
3) Что такое инкапсуляция? Какие ключевые слова С++ относятся к инкапсуляции?
4) Определите понятие объекта. Перечислите варианты отношений между объектами, в чем состоит взаимосвязь поведения и состояния объекта?
5) Определите понятие класса. Перечислите варианты отношений между классами. Обозначьте взаимосвязь между этими отношениями, если она существует. Какое из отношений является наиболее сильным по степени зависимости между классами?
6) Что такое абстрактные классы и интерфейсы? Для чего они нужны в ООП?
7) В приведенном классе перечислите (отметьте) переменные и методы, обеспечивающие состояние и поведение объектов класса Detail:

```
class Detail {
const int serial;
Detail *pcomponents;
double weight;
static const int ARTICLE;
  protected:
Detail();
Detail(const Detail&);
  public:
        Detail (int);
        static void performAction();
        int getSerial() const;
        double getWeight() const;
        void doSomething();
        ~Detail();
};
```

8) В приведенном примере класса MobilePhone, отметьте, какие методы являются конструкторами, деструкторами, модификаторами и селекторами.

```
class MobilePhone {
//…
public:
MobilePhone (long networkId = 0);
        MobilePhone (const MobilePhone &);
        const MobilePhone & operator=(const MobilePhone &);
  int getOwnNumber() const;
```

```
  int popLastCall();
        void redial();
        void placeCall(int number);
        friend ostream& operator<<(ostream&, const MobilePhone&);
  ~MobilePhone ();
        static Network* getAllKnownNetworks();
};
```

9) Перечислите примитивные типы языка C++. Что вы можете отметить об относительном размере (sizeof) этих типов по отношению друг к другу?

10) Напишите следующие объявления (переменных): указатель на булеву переменную, указатель на константный long, массив из 10 объектов string, указатель на указатель на символ, константный указатель на целое , ссылка на float, ссылка на константный объект типа string. Проинициализируйте эти переменные значениями, используя определения.

11) Напишите функцию, которая обменивает значения двух чисел типа double. В качестве типа аргументов воспользуйтесь указателями на double. Напишите аналогичную функцию, использующую ссылочные аргументы.

12) Перечислите классы памяти в программах написанных на языке C++.

13) Отметьте, в какой памяти будет выделено пространство под каждую из перечисленных переменных (ее содержимое):

```
struct Record {
        static const int MAX_VALUE=0x7FFFFFFF;                  //MAX_VALUE?
        int value;
        const int constValue;
        static double staticValue;
Record(): constValue(0) {}
        static void foo();
};
Record object;                          //object?

double Record::staticValue = 1.2;       //staticValue?

void Record::foo() {
  static int counter=1;          //counter?
        Record o;                       //o.value?        o.constValue?
        Record &r = o;                  //r.value?        r.constValue?
}

const double PI= 3.1415926;             //PI?
int main (int argc, char** argv) {     //argc?
  Record &ref = object;                 //ref?        ref.staticValue?
        string s[10];                   //s[1] ?
  Record *ptr = &ref;                   //ptr->value?
  int i =10;                            //i?
        Record *pobj = new Record[10];  //pobj?           *pobj?
        pobj[5]?
}
```

14) Что распечатает следующая программа на C++, какое место в программе Вам кажется потенциально опасным и почему?

```
#include <iostream>
using namespace std;

class A{
public:
  A(){
cout<<"+";
}
  A(const A& a){
cout<<"#";
  }
  ~A(){
cout<<"-";
}
};
class B{
  A a;
public:
  B(A tmp){
                cout <<"B";
a = tmp;
}
  A& getA() {
return a;
}
};

A global;
int main(int argc, char **argv) {
  cout<<"begin";
  A *pa = new A();
A local(*pa);
  B b(global);
  local = b.getA();
  cout<<"finish";
}
```

15) Что распечатает следующая программа на C++ (в предположении что calloc отработает успешно), дополнительно укажите что в этой программе на Ваш взгляд неправильно/некрасиво и почему?:

```
#include <iostream>
#include <stdlib>
using namespace std;
class A{
   static int counter;
   int i;
public:
   A() {
      i = ++counter;
```

```
    }
    ~A() {
       counter--;
    }
    friend ostream& operator<<(ostream &, const A&);
};
ostream& operator<<(ostream & out, const A& a) {
    return out<<a.i;
}

int A::counter = 0;
A global;

int main(int argc, char **argv) {
   cout<<global<<endl;
   A *p2A = static_cast<A*> ( calloc(5, sizeof(A)) );
A localArray[5];
   A *pA = new A[3];
   cout<< p2A[2] << endl;
cout<< localArray[3] << endl;
   cout<< *pA << endl;
   delete[] pA;
   free (p2A);
   A local;
   cout<< local << endl;
}
```

16) Для класса Widget задайте констркутор (агументы определите самостоятельно), конструктор копии, деструктор и оператор присваивания, при условии что классы Detail и Specification полностью реализованы:

```
class Widget {
  private:
    const int serial;  //серийный номер
    double weight;
    Specification *pspec; //указатель на спецификацию (ассоциация)
    int detailsCount; //число компонент
    Detail *pdetail;  //компоненты (композиция)




};
```

2.1.2. Требования к портфолио

Портфолио должно содержать результаты участия в коллоквиуме (письменный тест) и 4-6 выполненных заданий по следующим темам:

- Раздельная компиляция и пространства имен, структурные средства языка C++
- Перегрузка функций, указатели на функции, перечисления
- Классы в языке C++

- Иерархии классов, наследование
- Обобщенное программирование

2.1.3. Форма и перечень вопросов экзаменационного билета

Форма экзаменационного билета

Таблица П1.3

Новосибирский государственный университет

Экзамен

Основы объектно-ориентированного программирования

наименование дисциплины

09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
Программная инженерия и компьютерные науки

наименование образовательной программы

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №

1. Вопрос из категории 1
2. Вопрос из категории 2

Составитель

_____ В.Ю. Рылов
(подпись)

Ответственный за образовательную программу

_____ А.А. Романенко
(подпись)

«____»_____20 г.

Перечень вопросов экзамена, структурированный по категориям, представлен в таблице П1.4

Категория, компетенции	Формулировка вопроса
Категория 1 – Основы объектного подхода	
ОПК -1.1	Эволюция методологий программирования. Парадигмы программирования
ОПК -1.1	Основные принципы объектного подхода. Абстрагирование
ОПК -1.1	Основные принципы объектного подхода. Инкапсуляция
ОПК -1.1	Основные принципы объектного подхода. Модульность
ОПК -1.1	Основные принципы объектного подхода. Иерархия
ОПК -1.1	Основные принципы объектного подхода. Типизация
ОПК -1.1	Объект с точки зрения ООП. Состояние. Поведение
ОПК -1.1	Объект с точки зрения ООП. Идентичность и жизненный цикл объектов
ОПК -1.1	Объект с точки зрения ООП. Взаимоотношения между объектами.

ОПК -1.1	Классы. Природа классов. Метамодель. Инстанцирование.
ОПК -1.1	Классы. Структура класса. Абстрактные классы и интерфейсы
ОПК -1.1	Классы. Принцип подстановки Лисковой. Принцип разделения интерфейсов
ОПК -1.1, ОПК-1.2	Классы. Средства UML для построения диаграмм классов
ОПК -1.1, ОПК-1.2	Классы. Отношения между классами. Ассоциация и агрегация
ОПК -1.1, ОПК-1.2	Классы. Иерархии классов. Зависимость
Категория 2 – Средства объектно-ориентированного программирования на C++	
ОПК-8.1	Модель памяти и структура программы. Классы памяти. Ссылки
ОПК-8.1	Средства абстракции C++. Структура класса. Статические члены и их инициализация
ОПК-8.1	Средства инкапсуляции C++. Инкапсуляция и наследование. Друзья
ОПК-8.1	Модульность, раздельная компиляция, пространства имен, using директива.
ОПК-8.1	Представление иерархических отношений. Наследование
ОПК-8.1	Представление иерархических отношений. Агрегация. Зависимость по времени жизни
ОПК-8.1	Правила преобразования типов в C++. Параметрический и виртуальный полиморфизм
ОПК-8.1	C++: средства реализации состояния объектов; реализация поведения
ОПК-8.1	Перегрузка операторов
ОПК-8.1	Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании
ОПК-8.1	Варианты реализации отношения клиент-сервер. Объекты при передаче параметров и возврате из методов
ОПК-8.1	Исключения в C++. Обработка исключений. Умные указатели
ОПК-8.1	Шаблоны классов и шаблоны функций. Специализация
ОПК-8.1	Основы STL. Структура и назначение. Контейнеры
ОПК-8.1	Основы STL. Аллокаторы и итераторы

Набор экзаменационных билетов формируется и утверждается в установленном порядке в начале учебного года при наличии контингента обучающихся, завершающих освоение дисциплины «Основы объектно-ориентированного программирования» в текущем учебном году.

3. Критерии оценки сформированности компетенций в рамках промежуточной аттестации по дисциплине

Таблица П1.5

Шифр компетенций	Структурные элементы оценочных средств	Показатель сформированности	Не сформирован («неудовлетворительно»)	Пороговый уровень («удовлетворительно»)	Базовый уровень («хорошо»)	Продвинутый уровень («отлично»)
ОПК-1	Портфолио, Экз. вопрос категории 1	ОПК-1.1 Знать:основы математики, физики, вычислительной техники и программирования	Не знает нотации языка моделирования UML в части диаграмм классов, диаграмм последовательности	Знает основные элементы нотации, может читать диаграммы	Знает все основные элементы языка UML для моделирования отношений между классами их назначение	Знает особенности применения элементов нотации UML
ОПК-1	Портфолио, Экз. вопрос категории 1	ОПК- 1.2 Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования	Не знает нотации, не умеет строить диаграммы UML для визуализации отношений между классами	Может представлять классы и их структуру, может представить диаграмму классов своей программы	Умеет декорировать (именовать, аннотировать, специфицировать мощность) отношения между классами, знает различия в обозначениях зависимости, ассоциации и агрегации и композиции	Уверенно строит диаграммы классов , решения практических задания портфолио снабжены диаграммами демонстрирующими структуру программы
ОПК-1	Портфолио	ОПК-1.3 Владеть: навыками теоретического и экспериментального исследования объектов профессиональной дея-	Не знает какие основные средства используются на разных платформах (Windows, Linux, Mac OS X). Не	Знает основные характеристики сред для разных платформ. Умеет использовать основные	Знает характеристики сред для разных платформ, Демонстрирует	Демонстрирует глубокие познания в достоинствах и недостатках основных

		тельности	умеет пользоваться средой разработки	функции для редактирования одномодульной программы и компиляции	уверенные знания функций одной из сред разработки. Умеет пользоваться средой разработки для реализации многомодульных приложений владеет средствами отладки	средств разработки для разных платформ, уровень поддержки ими современного стандарта языка. Умеет пользоваться средой разработки, разрабатывать многомодульные приложения, пользоваться полным спектром средств оптимизации и отладки
ОПК-8	Портфолио, Экз. вопрос категории 2	ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения	Имеет фрагментарные знания в области структурных средств разработки C++, не знает правил перегрузки операций и функций	Знает основные алгоритмические и структурные средства языка, основные типы данных и модель сборки	Демонстрирует уверенные знания основных элементов и структурных средств языка, типов данных правил разрешения параметрического полиморфизма	Демонстрирует углубленные знания основных алгоритмических, функциональных и процедурных средств, знает особенности оптимизации на уровне языка ассемблера
ОПК-8.2	Портфолио (сдача заданий)	ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность про-	не умеет писать тесты с целью верификации корректности реализованных программ с использованием библиотеки Google Test	Умеет разрабатывать простейшие тесты с использованием технологии Google Test	Уверенно пишет тесты с использованием технологии Google Test. Степень тестового по-	Пользуется продвинутыми средствами технологии Google Test, тестовое покрытие программ практи-

		граммы , интегрировать программные модули			крытия программ практического задания не менее 50%	ческих заданий не менее 75%
ОПК-8.3	Портфолио (сдача заданий)	ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы	Не умеет реализовать алгоритмы и определять собственные типы данных (классы)	Умеет реализовать основные алгоритмы с использованием простых средств языка, может определять несложные классы	Реализует алгоритмы, разрабатывает систему классов, может реализовать простые шаблоны и обобщенные алгоритмы	Владеет всеми средствами языка, может программировать сложные шаблоны с переменным числом аргументов

4. Критерии выставления оценок по результатам промежуточной аттестации по дисциплине

Результаты промежуточной аттестации в 3 семестре определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Оценка «отлично» соответствует продвинутому уровню сформированности компетенции.

Оценка «хорошо» соответствует базовому уровню сформированности компетенции.

Оценка «удовлетворительно» соответствует пороговому уровню сформированности компетенции.

Оценка «неудовлетворительно» выставляется, если хотя бы одна компетенция не сформирована.

Итоговая оценка результатов промежуточной аттестации выставляется по следующей формуле:

Итоговая Оценка = 0.4 • O_1 + 0.6 • O_2;

O_1 - итоговая оценка по компетенциям, не вынесенным экзамен,

O_2 - итоговая оценка по компетенциям, вынесенным на экзамен.

Оценки O_1 и O_2 представляют из себя соответствующие средние арифметические оценок по компетенциям не вынесенным и вынесенным соответственно на экзамен.

Лист актуализации фонда оценочных средств
промежуточной аттестации по дисциплине
«Основы объектно-ориентированного программирования»

№	Характеристика внесенных изменений (с указанием пунктов документа)	Дата и № протокола Ученого совета ФИТ	Подпись ответственного
1.	Актуализирован на 2020-2021 уч.год	22.07.2020 №77