

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра теоретической и прикладной информатики
Кафедра прикладной математики

“УТВЕРЖДАЮ”
ДЕКАН ФПИИ
д.т.н. Тимофеев В. С.
“ ____ ” _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Объектно-ориентированное программирование

Образовательная программа: 01.03.02 Прикладная математика и информатика, профиль:
Компьютерное моделирование и информационные технологии

Курс: 2, семестр : 3

Факультет прикладной математики и информатики,

		Семестр
№	Вид деятельности	3
1	Всего зачетных единиц (кредитов)	4
2	Всего часов	144
3	Всего занятий в контактной форме, час.	68
4	Лекции, час.	18
5	Практические занятия, час.	0
6	Лабораторные занятия, час.	36
7	из них в активной и интерактивной форме, час.	11
8	Аттестация, час.	2
9	Консультации, час.	12
10	Самостоятельная работа, час.	76
11	Виды самостоятельной работы (курсовой проект, курсовая работа, РГЗ, подготовка к контрольной работе)	РГЗ
12	Вид аттестации	ДЗ

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 01.03.02 Прикладная математика и информатика

ФГОС введен в действие приказом №228 от 12.03.2015 г. , дата утверждения: 14.04.2015 г.

Место дисциплины в структуре учебного плана: Блок 1, вариативная

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 01.03.02 Прикладная математика и информатика

Рабочая программа обсуждена на заседании кафедры

ТПИ, протокол заседания кафедры №4 от 20.06.2017

ПМт, протокол заседания кафедры №6 от 20.06.2017

Утверждена на совете факультета прикладной математики и информатики, протокол № 6 от 21.06.2017

Программу разработал:

профессор, д.т.н. Лисицин Д. В.

Заведующий кафедрой:

доцент, д.т.н. Чубич В. М.

профессор, д.т.н. Соловейчик Ю. Г.

Ответственный за образовательную программу:

заведующий кафедрой Соловейчик Ю. Г.

1. Внешние требования

Таблица 1.1

Компетенция ФГОС: ОПК.3 способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям; в части следующих результатов обучения:
у3. уметь выбирать язык программирования, наиболее эффективный для решаемой задачи
у4. владеть средствами и технологиями разработки программного обеспечения, в том числе современными языками программирования, стандартными алгоритмическими решениями, сетевыми технологиями
Компетенция ФГОС: ПК.3 способность критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности; в части следующих результатов обучения:
з2. знать основные технологии программирования
Компетенция ФГОС: ПК.7 способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения; в части следующих результатов обучения:
з2. знать основные принципы ООП

2. Требования НГТУ к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий
Объектно-ориентированное программирование	
ПК.3.з2 знать основные технологии программирования	
1.Иметь представление о новейших направлениях в области создания технологий программирования и о роли объектно-ориентированного подхода среди них	Лекции; Лабораторные работы; Самостоятельная работа
ПК.7.з2 знать основные принципы ООП	
2.Иметь представление о множестве задач, решаемых с использованием объектно-ориентированного подхода	Лекции; Лабораторные работы; Самостоятельная работа
ПК.3.з2 знать основные технологии программирования	
3.Знать основные причины сложности программного обеспечения и пути ее преодоления с помощью новых технологий программирования	Лекции; Лабораторные работы; Самостоятельная работа
4.Знать основы тестирования объектно-ориентированных программ	Лекции; Лабораторные работы; Самостоятельная работа
ПК.7.з2 знать основные принципы ООП	
5.Знать основные принципы, лежащие в основе объектно-ориентированного подхода	Лекции; Лабораторные работы; Самостоятельная работа
6.Знать понятие и характеристики объекта, типы отношений между объектами	Лекции; Лабораторные работы; Самостоятельная работа
7.Знать понятие класса и виды отношений между классами, понятия переменных и операций класса, категорий классов и интерфейса	Лекции; Лабораторные работы; Самостоятельная работа
8.Знать синтаксис и семантику языка программирования C++	Лекции; Лабораторные работы; Самостоятельная работа
9.Знать элементы языка визуального моделирования UML	Лекции; Самостоятельная работа
10.Знать основы объектно-ориентированного анализа	Лекции; Самостоятельная работа
ОПК.3.у4 владеть средствами и технологиями разработки программного обеспечения, в том числе современными языками программирования, стандартными алгоритмическими решениями, сетевыми технологиями	

11. Уметь программировать на языке C++	Лабораторные работы; Самостоятельная работа
ОПК.3.у3 уметь выбирать язык программирования, наиболее эффективный для решаемой задачи	
12. Уметь программировать с использованием принципов и средств объектно-ориентированного подхода	Лабораторные работы; Самостоятельная работа
ОПК.3.у4 владеть средствами и технологиями разработки программного обеспечения, в том числе современными языками программирования, стандартными алгоритмическими решениями, сетевыми технологиями	
13. Уметь пользоваться средствами языков C++ и UML и систем Visual C++ и Enterprise Architect для выражения проектных решений	Лабораторные работы; Самостоятельная работа

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лекций	Активные формы, час.	Часы	Ссылки на результаты обучения
Семестр: 3			
Дидактическая единица: Сложность программного обеспечения.			
1. Эволюция технологий программирования. Основные идеи и преимущества использования объектно-ориентированного подхода к разработке программного обеспечения. Объектно-ориентированный подход как ответ на растущую сложность программного обеспечения. Сложные промышленные программные продукты. Примеры и характеристики задач, решаемых с использованием объектно-ориентированного подхода. Причины сложности, присущей программному обеспечению. Общие признаки сложных систем.	0	3	1, 2, 3
Дидактическая единица: Объектная модель.			
2. Объектная модель - принципы, лежащие в основе объектно-ориентированного подхода. Абстрагирование: абстракция, барьер и уровни абстракции, контрактная модель программирования, инварианты, обработка исключительных ситуаций. Инкапсуляция: интерфейс и реализация, инструменты скрытия реализации. Модульность: модуль, разделение описание-реализация, принципы, приемы и правила разбиения системы на модули, подсистемы. Иерархичность: иерархии типа "является" и "имеет", одиночное и множественное наследование. Типизация: согласование типов, полиморфизм и его виды, операторы приведения. Параллелизм: поток управления, активные и неактивные объекты. Сохраняемость: сохранение объектов в пространстве и времени.	0	3	5, 8
Дидактическая единица: Объекты.			
3. Объекты. Состояние объекта. Поведение объекта: операции (модификатор, селектор, конструктор, деструктор), конструктор с одним аргументом, инициализаторы конструктора, свободные подпрограммы. Идентичность объекта: проблема структурной зависимости, конструктор копирования, присваивание и равенство объектов. Отношения между объектами: связи и агрегация.	0	2	6, 8
Дидактическая единица: Классы.			

4. Классы. Виды отношений между классами. Ассоциация: мощность ассоциации, рефлексивная ассоциация. Агрегация и композитная агрегация. Зависимость (использование). Наследование: построение наследственных иерархий; конкретные, абстрактные, корневые и листовые классы; операции и методы класса; виртуальные и чисто виртуальные функции; защищенная часть класса. Наследование и типизация: открытое, закрытое и защищенное наследование; чистый полиморфизм, динамическое определение типа, повышающее и понижающее приведение. Множественное наследование: конфликт имен между суперклассами, повторное наследование, виртуальное наследование. Инстанцирование: параметризованный класс, параметрический полиморфизм. Статические элементы класса (переменные и операции класса), утилиты. Интерфейсы. Категории классов и пространства имен.	0	4	7, 8
Дидактическая единица: Основные конструкции языка UML.			
5. Основные конструкции языка визуального моделирования UML: диаграмма классов, диаграмма объектов, диаграммы последовательностей и коммуникации.	0	3	9
Дидактическая единица: Современные технологии программирования.			
6. Визуальное, компонентное программирование.	0	1	1, 2, 3, 5
Дидактическая единица: Основы тестирования объектно-ориентированных программ.			
7. Основы тестирования объектно-ориентированных программ.	0	1	4
Дидактическая единица: Основы объектно-ориентированного анализа.			
8. Основы объектно-ориентированного анализа.	0	1	10, 2, 3

Таблица 3.2

Темы лабораторных работ	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 3				
Дидактическая единица: Сложность программного обеспечения.				
1. Программирование графики.	2	8	11, 13, 3, 4, 8	Изучить необходимые средства для написания программ на языке C/C++, работающих с графикой; разработать функции, рисующие на экране геометрические фигуры, и программу, демонстрирующую выполнение указанных функций, включая ввод параметров фигур, параметров рисуемых линий и закраски; активная форма проведения занятий: дискуссия на тему "Возможности программирования графики в операционной системе Windows"
Дидактическая единица: Объектная модель.				

2. Классы.	2	6	1, 11, 12, 13, 2, 3, 4, 5, 8	Изучить средства работы с классами на языке С++, средства разработки многофайловых программ, средства работы с исключительными ситуациями; модифицировать программу, разработанную в лабораторной работе №1, так чтобы в ней был определен класс, реализующий понятие геометрической фигуры в графической системе; активная форма проведения занятий: дискуссия на тему "Принципы объектной модели".
Дидактическая единица: Объекты.				
3. Объекты.	3	8	1, 11, 12, 13, 2, 3, 4, 6, 8	Изучить понятие объекта, средства его создания и уничтожения; разработать программу, в которой был бы определен класс-контейнер для объектов класса, разработанного в лабораторной работе №2; класс включает конструктор, деструктор, функции для помещения объектов в контейнер, их удаления, поиска в контейнере, вывода содержимого контейнера; активная форма проведения занятий: дискуссия на тему "Объекты: состояние, поведение, идентичность, отношения между объектами".
Дидактическая единица: Классы.				
4. Отношения между классами.	2	6	1, 11, 12, 13, 2, 3, 4, 7, 8	Изучить реализацию на языке С++ отношений между классами: агрегации, наследования, зависимости; путем модификации программы, разработанной в лабораторной работе № 2, разработать программу, такую чтобы в ней были определены несколько классов, реализующих понятие геометрической фигуры в графической системе и имеющих между собой изучаемые отношения; активная форма проведения занятий: дискуссия на тему "Отношения между классами: ассоциация, агрегация, зависимость, наследование".

5. Шаблоны.	2	8	11, 12, 13, 4, 7, 8	Изучить понятия шаблона, инстанцирования; путем модификации программ, разработанных в лабораторных работах № 3, 4, разработать шаблон контейнера для хранения объектов классов, реализующих геометрические фигуры; активная форма проведения занятий: дискуссия на тему "Шаблоны, статические элементы класса, интерфейсы, категории классов и пространства имен"
-------------	---	---	---------------------	---

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 3				
1	РГЗ	1, 10, 11, 12, 13, 2, 3, 5, 7, 8, 9	9	3
При выполнении РГЗ студент может выбрать одну из двух тем: 1. Объектно-ориентированный анализ. 2. Создание диаграммы классов и прототипа кода на языке C++ в среде Enterprise Architect. Выполнение РГЗ соответствует дидактическим единицам "Основные конструкции языка UML", "Современные технологии программирования", "Основы объектно-ориентированного анализа" : Объектно-ориентированное программирование : методические указания к выполнению расчетно-графического задания для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2015. - 29, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000219940 Лисицин Д. В. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Д. В. Лисицин ; Новосиб. гос. техн. ун-т. - Новосибирск, [2013]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000179323. - Загл. с экрана.				
2	РГЗ	1, 10, 11, 12, 13, 2, 3, 5, 7, 8, 9	1	1
Защита расчетно-графического задания путем собеседования по выполненному заданию и/или ответов на контрольные вопросы (тема 1), ответов на контрольные вопросы (тема 2). Подготовка к защите РГЗ (тема 1) производится по отчету и/или по контрольным вопросам. Подготовка к защите РГЗ (тема 2) производится по контрольным вопросам. : Объектно-ориентированное программирование : методические указания к выполнению расчетно-графического задания для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2015. - 29, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000219940				
3	подготовка к выполнению лабораторных работ	1, 11, 12, 13, 2, 3, 4, 5, 6, 7, 8	50	0
изучение соответствующего теоретического материала и методических указаний, анализ задачи, проектирование и разработка необходимого программного обеспечения, разработка системы тестов, оформление отчета: Объектно-ориентированное программирование : методические указания к лабораторным работам для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2010. - 40, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000149206 Лисицин Д. В. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Д. В. Лисицин ; Новосиб. гос. техн. ун-т. - Новосибирск, [2013]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000179323. - Загл. с экрана.				
4	подготовка к лекциям	1, 10, 2, 3, 4, 5, 6, 7, 8, 9	4	6

изучение материала предшествующих лекций: Лисицин Д. В. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Д. В. Лисицин ; Новосиб. гос. техн. ун-т. - Новосибирск, [2013]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000179323. - Загл. с экрана.

5	Подготовка к аттестации	10, 11, 12, 13, 2, 5, 6, 7, 8, 9	12	2
Изучение теоретического материала и типового варианта комплексного задания для итоговой аттестации: Лисицин Д. В. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Д. В. Лисицин ; Новосиб. гос. техн. ун-т. - Новосибирск, [2013]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000179323 . - Загл. с экрана.				

5. Технология обучения

Для организации и контроля самостоятельной работы обучающихся, а также проведения консультаций применяются информационно-коммуникационные технологии (табл. 5.1).

Таблица 5.1

Деятельность	Информационно-коммуникационные технологии
Информирование	Личный типовой сайт: http://ciu.nstu.ru/kaf/persons/842/about
Размещение учебных материалов	ЭБС

Таблица 5.2

Активные и интерактивные формы проведения занятий

№	Наименование активных форм	Коды формируемых компетенций
1	Дискуссия	ПК.3; ПК.7;
Формируемые умения: з2. знать основные принципы ООП; з2. знать основные технологии программирования		
Краткое описание применения: Беседа студента с преподавателем на предложенные преподавателем темы (контрольные вопросы).		

6. Правила аттестации обучающихся по учебной дисциплине

Для аттестации обучающихся по дисциплине используется балльно-рейтинговая система (БРС), позволяющая выставять оценки по традиционной шкале и 15-уровневой ECTS.

Краткая информация о БРС приведена в табл. 6.1.

Подробная информация о БРС приводится в Приложении №4 к рабочей программе.

Таблица 6.1

Оцениваемые виды деятельности обучающихся	Мин. балл	Максимальный балл
Семестр: 3		
<i>Лекция:</i>	0	8
<i>Лабораторная №1: Выполнение</i>	4	6
Контролирующие материалы (список заданий) приводятся в "Объектно-ориентированное программирование : методические указания к лабораторным работам для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2010. - 40, [3] с. : ил. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000149206 "		
<i>Лабораторная №1: Защита</i>	3	6
Контролирующие материалы (список вопросов) приводятся в "Объектно-ориентированное программирование : методические указания к лабораторным работам для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2010. - 40, [3] с. : ил. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000149206 "		

Лабораторная №2: Выполнение	4	6
Контролирующие материалы (список заданий) приводятся в "Объектно-ориентированное программирование : методические указания к лабораторным работам для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2010. - 40, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000149206 "		
Лабораторная №2: Защита	3	6
Контролирующие материалы (список вопросов) приводятся в "Объектно-ориентированное программирование : методические указания к лабораторным работам для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2010. - 40, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000149206 "		
Лабораторная №3: Выполнение	4	6
Контролирующие материалы (список заданий) приводятся в "Объектно-ориентированное программирование : методические указания к лабораторным работам для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2010. - 40, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000149206 "		
Лабораторная №3: Защита	3	6
Контролирующие материалы (список вопросов) приводятся в "Объектно-ориентированное программирование : методические указания к лабораторным работам для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2010. - 40, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000149206 "		
Лабораторная №4: Выполнение	4	6
Контролирующие материалы (список заданий) приводятся в "Объектно-ориентированное программирование : методические указания к лабораторным работам для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2010. - 40, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000149206 "		
Лабораторная №4: Защита	3	6
Контролирующие материалы (список вопросов) приводятся в "Объектно-ориентированное программирование : методические указания к лабораторным работам для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2010. - 40, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000149206 "		
Лабораторная №5: Выполнение	4	6
Контролирующие материалы (список заданий) приводятся в "Объектно-ориентированное программирование : методические указания к лабораторным работам для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2010. - 40, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000149206 "		
Лабораторная №5: Защита	3	6
Контролирующие материалы (список вопросов) приводятся в "Объектно-ориентированное программирование : методические указания к лабораторным работам для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2010. - 40, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000149206 "		
РГЗ: Выполнение	4	6
Контролирующие материалы (список заданий) приводятся в "Объектно-ориентированное программирование : методические указания к выполнению расчетно-графического задания для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2015. - 29, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000219940 "		
РГЗ: Защита	3	6
Контролирующие материалы (список вопросов) приводятся в "Объектно-ориентированное программирование : методические указания к выполнению расчетно-графического задания для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2015. - 29, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000219940 "		
Зачет:	8	20
Контролирующие материалы (Комплект заданий) приводятся в "Лисицин Д. В. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Д. В. Лисицин ; Новосиб. гос. техн. ун-т. - Новосибирск, [2013]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000179323 . - Загл. с экрана."		

В таблице 6.2 представлено соответствие форм контроля заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.2

Коды компетенций ФГОС	Результаты обучения	Формы контроля		
		Защита ЛР	Защита РГЗ	Зачет
ОПК.3	у3. уметь выбирать язык программирования, наиболее эффективный для решаемой задачи	+	+	+
	у4. владеть средствами и технологиями разработки программного обеспечения, в том числе современными языками программирования, стандартными алгоритмическими решениями, сетевыми технологиями	+	+	+
ПК.3	з2. знать основные технологии программирования	+	+	+
ПК.7	з2. знать основные принципы ООП	+	+	+

Фонд оценочных средств по дисциплине представлен в приложении № 1 к рабочей программе.

7. Литература

Основная литература

1. Лисицин Д. В. Объектно-ориентированное программирование : [конспект лекций] / Д. В. Лисицин ; Новосиб. гос. техн. ун-т, Фак. приклад. мат. и информатики. - Новосибирск, 2010. - 86, [1] с. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000142706
2. Хабибуллин И. . Программирование на языке высокого уровня C/C++ : учебное пособие для вузов по направлению 654600 "Информатика и вычислительная техника" / И. Ш. Хабибуллин. - СПб, 2006. - 485 с. : ил.

Дополнительная литература

1. Буч Г. Язык UML : руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон ; [пер. с англ. Н. Мухина]. - М., 2007. - 493 с. : ил.

Интернет-ресурсы

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>
2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
4. ЭБС "Znaniy.com" : <http://znaniy.com/>
5. :

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

1. Объектно-ориентированное программирование : методические указания к лабораторным работам для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2010. - 40, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000149206
2. Объектно-ориентированное программирование : методические указания к выполнению расчетно-графического задания для 2 курса ФПМИ / Новосиб. гос. техн. ун-т ; [сост. Д. В. Лисицин]. - Новосибирск, 2015. - 29, [3] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000219940
3. Лисицин Д. В. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Д. В. Лисицин ; Новосиб. гос. техн. ун-т. - Новосибирск, [2013]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000179323. - Загл. с экрана.

8.2 Специализированное программное обеспечение

- 1 Microsoft Visual C++
- 2 Enterprise Architect

9. Материально-техническое обеспечение

Презентационное оборудование

№	Наименование	Назначение
1	Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления)	Для сопровождения лекций

Компьютерный класс

№	Наименование	Назначение
1	Компьютерный класс (Компьютеры объединены в локальную сеть с выходом в Internet)	Выполнение лабораторных работ и РГЗ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра прикладной математики
Кафедра теоретической и прикладной информатики

“УТВЕРЖДАЮ”
ДЕКАН ФПИ
д.т.н., доцент В.С. Тимофеев
“ ” _____ Г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Объектно-ориентированное программирование

Образовательная программа: 01.03.02 Прикладная математика и информатика, профиль:
Компьютерное моделирование и информационные технологии

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине Объектно-ориентированное программирование приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОПК.3 способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	у3. уметь выбирать язык программирования, наиболее эффективный для решаемой задачи	Классы. Объекты. Отношения между классами. Шаблоны.	Отчет по лабораторной работе № 2, контрольные вопросы 1–8, Отчет по лабораторной работе № 3, контрольные вопросы 1–13, Отчет по лабораторной работе № 4, контрольные вопросы 1–6, Отчет по лабораторной работе № 5, контрольные вопросы 1–10, РГЗ, тема 1: задания 1-6, тема 2: задания 1-4	Зачет, задачи 1-6
ОПК.3	у4. владеть средствами и технологиями разработки программного обеспечения, в том числе современными языками программирования, стандартными алгоритмическими решениями, сетевыми технологиями	Классы. Объекты. Отношения между классами. Программирование графики. Шаблоны.	Отчет по лабораторной работе № 1, контрольные вопросы 1–14, Отчет по лабораторной работе № 2, контрольные вопросы 1–8, Отчет по лабораторной работе № 3, контрольные вопросы 1–13, Отчет по лабораторной работе № 4, контрольные вопросы 1–6, Отчет по лабораторной работе № 5, контрольные вопросы 1–10, РГЗ, тема 1: задания 1-6, тема	Зачет, задачи 1-6

			2: задания 1-4	
ПК.3/НИ способность критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности	32. знать основные технологии программирования	Визуальное, компонентное программирование. Классы. Объекты. Основы объектно-ориентированного анализа. Основы тестирования объектно-ориентированных программ. Отношения между классами. Программирование графики. Шаблоны. Эволюция технологий программирования. Основные идеи и преимущества использования объектно-ориентированного подхода к разработке программного обеспечения. Объектно-ориентированный подход как ответ на растущую сложность программного обеспечения. Сложные промышленные программные продукты. Примеры и характеристики задач, решаемых с использованием объектно-ориентированного подхода. Причины сложности, присущей программному обеспечению. Общие признаки сложных систем.	Отчет по лабораторной работе № 1, контрольные вопросы 1–14, Отчет по лабораторной работе № 2, контрольные вопросы 1–8, Отчет по лабораторной работе № 3, контрольные вопросы 1–13, Отчет по лабораторной работе № 4, контрольные вопросы 1–6, Отчет по лабораторной работе № 5, контрольные вопросы 1–10, РГЗ, тема 1: задания 1-6, тема 2: задания 1-4	Зачет, задачи 1-6
ПК.7/ППр способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	32. знать основные принципы ООП	Визуальное, компонентное программирование. Классы. Классы. Виды отношений между классами. Ассоциация: мощность ассоциации, рефлексивная ассоциация. Агрегация и композитная агрегация. Зависимость (использование). Наследование: построение наследственных иерархий; конкретные, абстрактные, корневые и листовые классы; операции и методы класса; виртуальные и чисто виртуальные функции; защищенная часть класса. Наследование и типизация: открытое, закрытое и защищенное наследование; чистый полиморфизм, динамическое определение типа, повышающее и понижающее приведение. Множественное наследование: конфликт имен между суперклассами, повторное наследование, виртуальное наследование. Инстанцирование: параметризованный класс, параметрический полиморфизм. Статические элементы класса (переменные	Отчет по лабораторной работе № 1, контрольные вопросы 1–14, Отчет по лабораторной работе № 2, контрольные вопросы 1–8, Отчет по лабораторной работе № 3, контрольные вопросы 1–13, Отчет по лабораторной работе № 4, контрольные вопросы 1–6, Отчет по лабораторной работе № 5, контрольные вопросы 1–10, РГЗ, тема 1: задания 1-6, тема 2: задания 1-4	Зачет, задачи 1-7

		и операции класса), утилиты. Интерфейсы. Категории классов и пространства имен. Объектная модель - принципы, лежащие в основе объектно-ориентированного подхода. Абстрагирование: абстракция, барьер и уровни абстракции, контрактная модель программирования, инварианты, обработка исключительных ситуаций. Инкапсуляция: интерфейс и реализация, инструменты скрытия реализации. Модульность: модуль, разделение описание-реализация, принципы, приемы и правила разбиения системы на модули, подсистемы. Иерархичность: иерархии типа "является" и "имеет", одиночное и множественное наследование. Типизация: согласование типов, полиморфизм и его виды, операторы приведения. Параллелизм: поток управления, активные и неактивные объекты. Сохраняемость: сохранение объектов в пространстве и времени. Объекты. Объекты. Состояние объекта. Поведение объекта: операции (модификатор, селектор, конструктор, деструктор), конструктор с одним аргументом, инициализаторы конструктора, свободные подпрограммы. Идентичность объекта: проблема структурной зависимости, конструктор копирования, присваивание и равенство объектов. Отношения между объектами: связи и агрегация. Основные конструкции языка визуального моделирования UML: диаграмма классов, диаграмма объектов, диаграммы последовательностей и коммуникации. Основы объектно-ориентированного анализа. Отношения между классами. Программирование графики. Шаблоны.		
--	--	---	--	--

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 3 семестре - в форме дифференцированного зачета, который направлен на оценку сформированности компетенций ОПК.3, ПК.3/НИ, ПК.7/ППр.

Зачет проводится в письменной форме, по комплектам заданий. Комплекты заданий составляются из

задач, соответствующим типовым задачам, приведенным в паспорте зачета, позволяющих оценить показатели сформированности соответствующих компетенций.

На подготовку к зачету отводится 14 часов.

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 3 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (РГЗ). Требования к выполнению РГЗ, состав и правила оценки сформулированы в паспорте РГЗ.

В 3 семестре этапами текущей аттестации являются лабораторные работы. Требования к их выполнению, состав и правила оценки сформулированы в паспорте лабораторных работ.

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.3, ПК.3/НИ, ПК.7/ППр, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра прикладной математики
Кафедра теоретической и прикладной информатики

Паспорт зачета

по дисциплине «Объектно-ориентированное программирование», 3 семестр

1. Методика оценки

Зачет проводится в письменной форме, по билету. Билет формируется в виде набора задач, соответствующих задачам типового задания.

Форма билета для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет ФПИИ

Билет № 1

к зачету по дисциплине «Объектно-ориентированное программирование»

Задача №1. (1 балл) Укажите ошибочные строки в функции main среди отмеченных буквами А, Б, В, Г, Д, Е.

```
class Stack1 {protected: int top; . . .};  
class Stack2 {public: int top; . . .};  
class Stack3 {private: int top; . . .};  
struct St {int top; . . .};  
class Stack4 {int top; . . .};  
main() {  
    Stack1 S1; Stack2 S2; Stack3 S3; St S; Stack4 S4; int top;  
    . . .  
    S1.top++; // А  
    . . .  
    S2.top++; // Б  
    . . .  
    S3.top++; // В  
    . . .  
    S.top++; // Г  
    . . .  
    S4.top++; // Д  
    . . .  
    top++; // Е  
    . . .  
}
```

Задача № 2. (1 балл) Укажите, какие из пар абстракций

- А. относятся к иерархии типа «является»,
- Б. относятся к иерархии типа «имеет»,
- В. не относятся к иерархиям типа «является» и «имеет».

1. «Компьютер» – «Процессор»
2. «Дерево» – «Береза»
3. «Растение» – «Корень»
4. «Пчела» – «Цветок»
5. «Автомобиль» – «Колесо»
6. «Цветы» – «Розы»

Задача № 3. (2 балла) Что напечатает программа.

```
class Point {public: static int count;...};
int Point :: count=0;
main() { Point P1, P2;
    ...
    P1.count =1; P2.count =2;
    printf("%d\n", P2.count);
    P1.count++;
    printf("%d\n", P1.count);
}
```

Задача № 4. (2 балла) Укажите ошибочные строки в функции main среди отмеченных буквами А, Б, В, Г, Д, Е.

```
class Transport // Транспортное средство
{public:      char* Registration_Number; // регистрационный номер
};
class Land_Transport: public Transport { // Сухопутное транспортное средство
public: int Shaft; // ведущая ось
};
class Water_Transport: public Transport { // Водное транспортное средство
public: int Displacement; // водоизмещение
};
class Amphibia: public Land_Transport, public Water_Transport {}; // Амфибия
main() {
    Amphibia amph;
    ...
    amph.Shaft; // А
    amph.Displacement; // Б
    amph.Water_Transport :: Displacement; // В
    amph.Registration_Number; // Г
    amph.Water_Transport :: Registration_Number; // Д
    amph.Transport :: Registration_Number; // Е
}
```

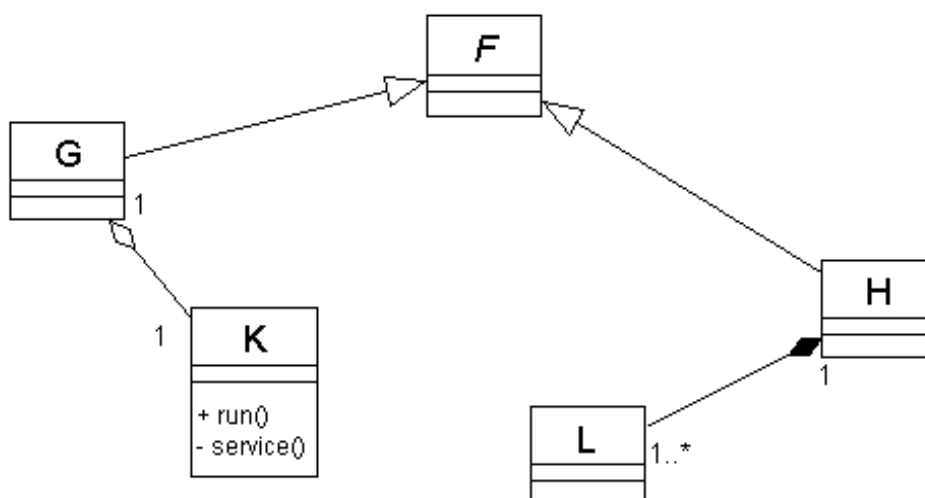
Задача № 5. (4 балла) Укажите ошибочные строки в функции main среди отмеченных буквами А, Б, В, Г, Д, Е.

```
class Shape { . . . } S;  
class Circle: public Shape { . . . } C;  
class SolidCircle: Circle { . . . } SC;  
class Square: Shape { . . . } SQ;  
main() {  
    . . .  
    S=C; // А  
    C=S; // Б  
    C=SC; // В  
    SC=C; // Г  
    S=SQ; // Д  
    SC=SQ; // Е  
}
```

Задача № 6. (4 балла) Что напечатает программа.

```
class Base { public:  
    void virtual info1 ( ) { printf("Base1\n"); }  
    void info2 ( ) { printf("Base2\n"); }  
};  
class Child : public Base { public:  
    void virtual info1 ( ) { printf("Child1\n"); }  
    void info2 ( ) { printf("Child2\n"); }  
};  
main() {  
    Base B, *PB;  
    Child C;  
    . . .  
    PB = &B; PB->info1();  
    PB = &C; PB->info1(); PB->info2();  
    B=C; B.info1();  
}
```

Задача № 7. (6 баллов) Составить словесное описание ситуации, изображенной на диаграмме классов.



Утверждаю: зав. кафедрой ____должность, ФИО
(подпись)
(дата)

2. Критерии оценки

Задачи 1, 2 оцениваются на 1 балл, если решены полностью, и на 0 баллов, если решение неполно или отсутствует. Задачи 3, 4 оцениваются на 2 балла, если решены полностью, и на 0 баллов, если решение неполно или отсутствует. Задачи 5, 6 оцениваются на 4 балла, если решены полностью, на 1 или 2 балла, если студент в процессе решения задач регулярно демонстрируется знание меньшего или большего количества правил решения задачи (но не всех), и на 0 баллов, если знание правил решения задач регулярно не демонстрируется или решение отсутствует. В задаче 7 правильное описание элемента диаграммы оценивается в 0.5 балла, а неправильное оценивается в 0 баллов, полученные баллы округляются до целого, при этом максимальный балл за решение задачи – 6.

- Зачет считается **не сданным**, если студент набрал менее 8 баллов
- Зачет считается сданным на **пороговом** уровне, если студент набрал 8-11 баллов
- Зачет считается сданным на **базовом** уровне, если студент набрал 12-16 баллов
- Зачет считается сданным на **продвинутом** уровне, если студент набрал 17-20 баллов

3. Шкала оценки

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Типовой вариант комплекта заданий к зачету по дисциплине «Объектно-ориентированное программирование» с ответами

Задача №1. (1 балл) Укажите ошибочные строки в функции main среди отмеченных буквами А, Б, В, Г, Д, Е.

```
class Stack1 {protected: int top; . . .};
class Stack2 {public: int top; . . .};
class Stack3 {private: int top; . . .};
struct St {int top; . . .};
class Stack4 {int top; . . .};
main() {
    Stack1 S1; Stack2 S2; Stack3 S3; St S; Stack4 S4; int top;
    . . .
    S1.top++; // А
    . . .
    S2.top++; // Б
    . . .
    S3.top++; // В
    . . .
    S.top++; // Г
    . . .
    S4.top++; // Д
    . . .
    top++; // Е
    . . .
}
```

Ответ: А, В, Д.

Задача № 2. (1 балл) Укажите, какие из пар абстракций
А. относятся к иерархии типа «является»,
Б. относятся к иерархии типа «имеет»,
В. не относятся к иерархиям типа «является» и «имеет».

1. «Компьютер» – «Процессор»
2. «Дерево» – «Береза»
3. «Растение» – «Корень»
4. «Пчела» – «Цветок»
5. «Автомобиль» – «Колесо»
6. «Цветы» – «Розы»

Ответ:

А. 2, 6

Б. 1, 3, 5

В. 4

Задача № 3. (2 балла) Что напечатает программа.

```
class Point {public: static int count;. . .};
int Point :: count=0;
main() { Point P1, P2;
    . . .
    P1.count =1; P2.count =2;
    printf("%d\n", P2.count);
}
```

```

        P1.count++;
        printf("%d\n", P1.count);
    }
Ответ:
2
3

```

Задача № 4. (2 балла) Укажите ошибочные строки в функции main среди отмеченных буквами А, Б, В, Г, Д, Е.

```

class Transport // Транспортное средство
{public:      char* Registration_Number; // регистрационный номер
};
class Land_Transport: public Transport { // Сухопутное транспортное средство
public: int Shaft; // ведущая ось
};
class Water_Transport: public Transport { // Водное транспортное средство
public: int Displacement; // водоизмещение
};
class Amphibia: public Land_Transport, public Water_Transport {}; // Амфибия
main() {
    Amphibia amph;
    ...
    amph.Shaft; // А
    amph.Displacement; // Б
    amph.Water_Transport :: Displacement; // В
    amph.Registration_Number; // Г
    amph.Water_Transport :: Registration_Number; // Д
    amph.Transport :: Registration_Number; // Е
}

```

Ответ: Г, Е.

Задача № 5. (4 балла) Укажите ошибочные строки в функции main среди отмеченных буквами А, Б, В, Г, Д, Е.

```

class Shape { . . . } S;
class Circle: public Shape { . . . } C;
class SolidCircle: Circle { . . . } SC;
class Square: Shape { . . . } SQ;
main() {
    ...
    S=C; // А
    C=S; // Б
    C=SC; // В
    SC=C; // Г
    S=SQ; // Д
    SC=SQ; // Е
}

```

Ответ: Б, В, Г, Д, Е.

Задача № 6. (4 балла) Что напечатает программа.

```

class Base { public:

```

```

void virtual info1 ( ){printf("Base1\n");}
void info2 ( ){printf("Base2\n");}
};
class Child : public Base { public:
void virtual info1 ( ){printf("Child1\n");}
void info2 ( ){printf("Child2\n");}
};
main() {
Base B, *PB;
Child C;

...
PB = &B; PB->info1();
PB = &C; PB->info1(); PB->info2( );
B=C; B.info1();
}

```

Ответ:

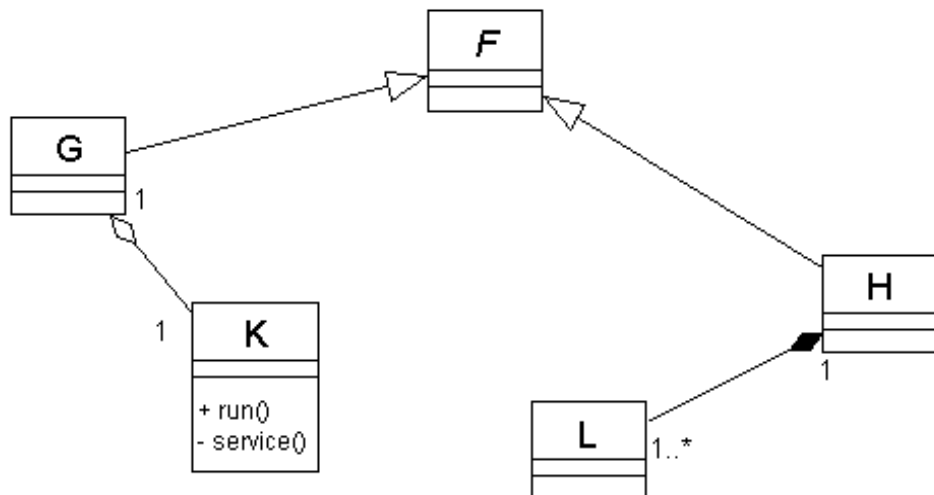
Base1

Child1

Base2

Base1

Задача № 7. (6 баллов) Составить словесное описание ситуации, изображенной на диаграмме классов.



Ответ. На диаграмме изображен абстрактный класс F и два его потомка: класс G и класс H. Оба класса-потомка являются агрегатами. Каждый объект класса G содержит один объект класса K. Каждый объект класса H содержит не менее одного объекта класса L (композиционная агрегация). Класс K имеет две операции: открытую – run() и закрытую service().

Паспорт расчетно-графического задания

по дисциплине «Объектно-ориентированное программирование», 3 семестр

1. Методика оценки

При выполнении РГЗ студент может выбрать одну из двух тем:

1. Объектно-ориентированный анализ.
2. Создание диаграммы классов и прототипа кода на языке C++ в среде Enterprise Architect.

Тема 1: Объектно-ориентированный анализ.

Задание 1. Изучить основные методы и этапы объектно-ориентированного анализа, назначение и основные элементы диаграмм классов, последовательностей и коммуникации языка UML. Ознакомиться с технологией разработки диаграмм и создания прототипа кода классов на языке C++ в системе Enterprise Architect.

Задание 2. Для выбранной предметной области выявить ключевые абстракции, составляющие словарь предметной области. На основе выявленных ключевых абстракций разработать систему классов и объектов, описывающих предметную область. Специфицировать атрибуты и операции классов, а также отношения, существующие между классами. Упорядочить классы в иерархии типа «является» (не менее одной) и иерархии типа «имеет». Представить разработанные классы в виде диаграмм классов языка UML.

Задание 3. Описать наиболее важные аспекты поведения системы посредством разработки нескольких сценариев, отражающих взаимодействие ряда объектов различных классов. Представить сценарии в виде диаграмм последовательностей или коммуникации языка UML.

Задание 4. Реализовать разработанные диаграммы классов в системе Enterprise Architect. Диаграммы последовательностей или коммуникации также можно реализовать в системе Enterprise Architect. Сгенерировать прототип кода средствами Enterprise Architect.

Задание 5. Оформить отчет, содержащий

- описание предметной области, словарь предметной области;
- спецификации разработанных классов и их назначение, диаграммы классов;
- словесное описание сценариев и их роль в моделировании предметной области, диаграммы последовательностей или коммуникации;
- прототип кода, сгенерированный в Enterprise Architect.

Задание 6. Защитить расчетно-графическое задание путем собеседования по выполненному проекту и/или ответов на контрольные вопросы.

Тема 2: Создание диаграммы классов и прототипа кода в системе Enterprise Architect.

Задание 1. Изучить назначение и основные элементы диаграммы классов языка UML. Ознакомиться с технологией разработки диаграмм классов и создания прототипа кода на языке C++ в системе Enterprise Architect.

Задание 2. Составить диаграмму классов графической системы, разработанной в лабораторных работах № 4, 5, и произвести автоматическую генерацию прототипа кода разработанных классов.

2.1. Поместить на диаграмму классы геометрических фигур и шаблон контейнера.

2.2. Специфицировать атрибуты и операции классов, а также отношения, существующие между классами.

2.3. Произвести автоматическую генерацию прототипа кода разработанных классов.

2.4. Сгенерированный код сравнить с кодом, полученным в лабораторных работах № 4, 5. В случае необходимости внести исправления в диаграмму классов и повторить генерацию кода.

Задание 3. Оформить отчет, содержащий постановку задачи, разработанную диаграмму классов и текст прототипа кода.

Задание 4. Защитить расчетно-графическое задание, ответив на вопросы преподавателя.

Список контрольных вопросов.

1. Диаграмма классов.

а. Как изображается класс? Какая информация указывается на его значке?

б. Каков формат спецификации атрибутов класса? Как обозначается на диаграмме видимость и кратность элементов?

в. Каков формат спецификации операций класса?

г. Как изображаются на диаграмме статические элементы и утилиты?

д. Как изображается на диаграмме интерфейс? Как специфицируется на диаграмме отношения между интерфейсом и реализующим его классом?

е. Как специфицируется на диаграмме отношение ассоциации между классами?

ж. Как специфицируются на диаграмме отношения зависимости, наследования, агрегации, композиции?

з. Как специфицируются на диаграмме шаблоны, отношение инстанцирования?

и. Что такое вложенный класс и как он специфицируется на диаграмме?

2. Диаграмма объектов.

а. Что отражает диаграмма объектов?

б. Как изображается объект? Какая информация указывается на его значке? Как изображаются на диаграмме отношения между объектами?

3. Диаграмма последовательностей.

а. Как изображаются на диаграмме объекты и вызовы операций?

б. Как изображается на диаграмме линия жизни объекта?

в. Как изображается на диаграмме фокус управления?

г. Как изображаются на диаграмме циклы и ветвления?

4. Диаграмма коммуникации.

а. Как изображаются на диаграмме объекты и вызовы операций?

б. Как изображаются на диаграмме итерации?

в. Как изображаются на диаграмме условия?

2. Критерии оценки

- Работа считается **не выполненной**, если основы языка UML освоены частично и пробелы носят существенный характер, необходимые практические навыки работы с диаграммами классов в системе Enterprise Architect сформированы частично, не все предусмотренные задания выполнены, большинство заданий выполнены с ошибками, отчет оформлен с недостатками, оценка составляет 0-6 баллов.
- Работа считается выполненной **на пороговом** уровне, если основы языка UML освоены частично, но пробелы не носят существенного характера, необходимые

практические навыки работы с диаграммами классов в системе Enterprise Architect в основном сформированы, все предусмотренные задания выполнены, но некоторые задания выполнены с ошибками, отчет оформлен с недостатками, оценка составляет 7-8 балла

- Работа считается выполненной **на базовом** уровне, если основы языка UML освоены полностью, без пробелов, необходимые практические навыки работы с диаграммами классов в системе Enterprise Architect в основном сформированы, все предусмотренные задания выполнены, некоторые из выполненных заданий, возможно, содержат несущественные ошибки, отчет оформлен без недостатков, оценка составляет 9-10 баллов
- Работа считается выполненной **на продвинутом** уровне, если основы языка UML освоены полностью, без пробелов, необходимые практические навыки работы с диаграммами классов в системе Enterprise Architect сформированы, все предусмотренные задания выполнены с качеством, близким к максимальному, отчет оформлен без недостатков, оценка составляет 11-12 баллов

3. Шкала оценки

В общей оценке по дисциплине баллы за РГЗ учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Примерный перечень тем РГЗ по теме 1

1. Книжный магазин.
2. Магазин промышленных товаров.
3. Склад.
4. Центр заказов.
5. Служба доставки.
6. Почтовое отделение.
7. Управление персоналом в компании (отдел кадров).
8. Управление компанией (заводом).
9. Цент межугородних (международных) переговоров.
10. Аэропорт.
11. Железнодорожный вокзал.
12. Управление железной дорогой.
13. Сортировочная станция.
14. Автострада.
15. Автовокзал.
16. Автотранспортные услуги.
17. Экосистема.
18. Факультет ВУЗа.
19. Школа.
20. Спортивный клуб (организация и управление).
21. Библиотека.
22. Банк.
23. Процесс обучения какому-либо предмету (организация и управление).

Паспорт лабораторных работ

по дисциплине «Объектно-ориентированное программирование», 3 семестр

1. Методика оценки

Лабораторная работа № 1.

Задание 1. Изучить основные понятия и функции графического интерфейса операционной системы Microsoft Windows.

Задание 2. По предложенному преподавателем варианту разработать функции, рисующие следующие геометрические фигуры:

- незакрашенную фигуру (фигуру-контур),
- закрашенную фигуру,
- две вложенные одна в другую фигуры, внешняя фигура закрашена, за исключением пространства внутренней фигуры.

Разработать программу, демонстрирующую выполнение указанных функций. Ввод параметров фигур (координат и др.), параметров рисуемых линий и закрашки осуществлять из файлов (отдельно для каждого теста). Включить в программу проверки корректности данных (нулевой радиус окружности, нарушение неравенства треугольника и т.д.), в том числе проверки нахождения фигуры в пределах окна и вложенности двух фигур.

Задание 3. Подготовить текстовый файл с разработанной программой, оттранслировать, собрать и выполнить программу с учетом требований операционных систем и программных оболочек, в которых эта программа выполняется. При необходимости исправить ошибки и вновь повторить технологический процесс решения задачи.

Задание 4. Оформить отчет по лабораторной работе. Отчет должен содержать постановку задачи, алгоритм, описание разработанных функций, текст разработанной программы и результаты тестирования.

Задание 5. Защитить лабораторную работу, ответив на вопросы преподавателя.

Контрольные вопросы.

1. Что такое «контекст отображения», как его получить и освободить?
2. Какую систему координат имеет контекст отображения, как определить размер окна?
3. Что такое «перо», какие имеются встроенные перья, как выбрать перо в контекст отображения?
4. Как создать новое перо, какие его характеристики можно задать, как удалить перо?
5. Что такое «кисть», какие имеются встроенные кисти, как выбрать кисть в контекст отображения?
6. Как создать новую кисть, какие ее характеристики можно задать, как удалить кисть?
7. Как нарисовать точку, как узнать цвет точки?
8. Что такое текущая позиция пера, как ее установить и узнать?
9. Как нарисовать прямую линию?
10. Как нарисовать ломаную линию?

11. Как нарисовать прямоугольник?
12. Как нарисовать эллипс?
13. Как нарисовать многоугольник?
14. Как вывести в окно текст, как установить цвет букв?

Лабораторная работа № 2.

Задание 1. Ознакомиться со следующими вопросами и понятиями: определение типа данных пользователя с помощью конструкции `class`, открытая и закрытая части описания класса, определение набора функций класса, интерфейс и реализация класса, реализация понятия модульности в языке C++.

Задание 2. Модифицировать программу, разработанную в лабораторной работе №1, так чтобы в ней был определен класс, реализующий понятие геометрической фигуры в графической системе.

2.1. Определить ответственность класса. Учесть, что общение с пользователем (включая ввод данных с клавиатуры и вывод данных на экран, если их предполагается использовать), а также реакцию на возникающие при работе с функциями класса ошибки следует производить вне функций класса.

2.2. Определить атрибуты, необходимые для реализации класса. Поместить атрибуты в закрытую часть описания.

2.3. Определить функции, необходимые для реализации класса. Выделить интерфейс класса и поместить его в открытую часть описания.

Включить в разработанный класс функции:

- устанавливающие и изменяющие геометрические и графические характеристики фигуры (set-функции);
- возвращающие геометрические и графические характеристики фигуры (get-функции);
- рисующие фигуру на экране;
- изменяющие положение фигуры на экране;
- обеспечивающие сохранимость объекта: сохранение набора атрибутов объекта класса в файле и считывание его из файла (файлы для сохранения и считывания должны иметь один формат).

2.4 Обработку ошибок (нулевой радиус окружности, нарушение неравенства треугольника, другие ошибки задания фигуры, ошибки работы с графикой и др.) осуществлять с использованием механизма возбуждения и обработки исключительных ситуаций.

2.5. Разработать функцию, демонстрирующую поведение класса. Поместить реализацию класса в один файл, а демонстрационную функцию – в другой. Обеспечить необходимые межмодульные связи.

Задание 3. Подготовить текстовые файлы с разработанной программой, оттранслировать их, собрать и выполнить программу с учетом требований операционных систем и программных оболочек, в которых эта программа выполняется. При необходимости исправить ошибки и вновь повторить технологический процесс решения задачи.

Задание 4. Оформить отчет по лабораторной работе. Отчет должен содержать постановку задачи, описание разработанных классов (атрибутов и функций), алгоритм, текст разработанной программы и результаты тестирования.

Задание 5. Защитить лабораторную работу, ответив на вопросы преподавателя.

Контрольные вопросы.

1. Принцип абстрагирования:

- а. В чем заключается принцип абстрагирования?
- б. Что такое барьер абстракции, уровни абстракции?
- в. В чем заключается принцип наименьшего удивления?
- г. Что такое контрактная модель программирования?

- д. Что такое сигнатура операции?
- е. Что такое инвариант?
- 2. Исключения:
 - а. Как определить пробный блок и обработчики исключений?
 - б. Как осуществить возбуждение исключительной ситуации?
 - в. Как происходит обработка исключений?
- 3. Инкапсуляция:
 - а. В чем заключается принцип инкапсуляции?
 - б. В чем смысл разделения класса на интерфейс и реализацию?
 - в. Что такое класс? Как он определяется? Как определяются операции для класса?
 - г. Как определяется открытая и закрытая части тела класса?
 - д. Что такое друг класса?
- 4. Модульность:
 - а. Что такое модуль?
 - б. В чем заключается принцип модульности?
 - в. Модульность на языке C++.
 - г. Какие существуют приемы эффективного разделения программы на модули?
- 5. Иерархичность:
 - а. В чем заключается принцип иерархичности?
 - б. Что такое иерархия «является»?
 - в. Что такое иерархия «имеет»?
- 6. Типизация:
 - а. В чем заключается принцип типизации?
 - б. Что такое полиморфизм?
 - в. Что такое принудительное приведение?
 - г. Что такое перегрузка?
 - д. Какие операторы приведения вы знаете?
- 7. В чем заключается принцип параллелизма?
- 8. В чем заключается принцип сохраняемости?

Лабораторная работа № 3.

Задание 1. Ознакомиться с вопросами использования конструкторов и деструкторов для создания и уничтожения объектов.

Задание 2. По предложенному преподавателем варианту разработать программу на языке C++, в которой был бы определен класс-контейнер, т.е. класс объектов, служащих для хранения объектов класса, разработанного в предыдущей лабораторной работе. Среди различных типов контейнеров упомянем списки, стеки, очереди, деревья, таблицы. Контейнер должен быть реализован как динамическая структура данных. Разработать следующие функции-члены класса: конструктор, деструктор, функции для помещения объектов-фигур в контейнер, их возвращения (удаления), поиска в контейнере, «распечатки содержимого» контейнера – вывода информации о содержащихся в объекте-контейнере объектах и/или их графических образов, а также функции, обеспечивающие сохраняемость контейнера с использованием файла.

Реализацию класса-контейнера поместить в отдельный файл. Разработать функцию, демонстрирующую поведение объекта-контейнера с несколькими объектами-фигурами.

Задание 3. Подготовить текстовые файлы с разработанной программой, оттранслировать их, собрать и выполнить программу с учетом требований операционных систем и программных оболочек, в которых эта программа выполняется. При необходимости исправить ошибки и вновь повторить технологический процесс решения задачи.

Задание 4. Оформить отчет по лабораторной работе. Отчет должен содержать постановку задачи, описание разработанных классов, алгоритм, текст разработанной

программы и результаты тестирования.

Задание 5. Защитить лабораторную работу, ответив на вопросы преподавателя.

Контрольные вопросы.

1. Что такое состояние объекта, чем оно определяется?
2. Что такое поведение объекта, посредством чего оно реализуется?
3. Какие виды типичных операций над объектами вы можете назвать?
4. Каким образом могут создаваться объекты?
5. Для чего используется ключевое слово `explicit`?
6. Что такое инициализатор конструктора, в каких случаях его использование обязательно?
7. Что такое свободная подпрограмма?
8. Что такое идентичность объекта?
9. Что такое структурная зависимость, какие проблемы она порождает?
10. Что такое конструктор копирования, какие проблемы могут возникнуть при использовании конструктора копирования, предоставляемого по умолчанию?
11. Какие проблемы могут возникнуть при использовании оператора присваивания, предоставляемого по умолчанию?
12. Что такое связь?
13. Что такое агрегация (отношение между объектами)?

Лабораторная работа № 4.

Задание 1. Ознакомиться с отношениями между классами: агрегацией, наследованием, зависимостью, а также с механизмом позднего связывания.

Задание 2. Путем модификации программ, разработанных в лабораторных работах № 1, 2, разработать программу такую, чтобы в ней были определены несколько классов, реализующих понятие геометрической фигуры в графической системе:

- абстрактный класс «Фигура», содержащий чисто виртуальные функции;
- класс «Закрашенный», позволяющий задать кисть, ее параметры и, возможно, осуществить закраску;
- класс «Фигура-контур» – потомок класса «Фигура»;
- класс «Закрашенная фигура» – потомок класса «Фигура-контур», класс «Закрашенный» при этом использовать либо как второго родителя (множественное наследование), либо как часть класса «Закрашенная фигура» (агрегация);
- класс «Комбинированная фигура», реализующий две вложенные фигуры с закраской между ними.

Реализацию классов поместить в отдельный файл.

Разработать функцию, демонстрирующую поведение разработанных классов, включая демонстрацию механизма позднего связывания.

Задание 3. Подготовить текстовые файлы с разработанной программой, оттранслировать их, собрать и выполнить программу с учетом требований операционных систем и программных оболочек, в которых эта программа выполняется. При необходимости исправить ошибки и вновь повторить технологический процесс решения задачи.

Задание 4. Оформить отчет по лабораторной работе. Отчет должен содержать постановку задачи, описание разработанных классов, алгоритм, текст разработанной программы и результаты тестирования.

Задание 5. Защитить лабораторную работу, ответив на вопросы преподавателя.

Контрольные вопросы.

1. Ассоциация:
 - а. Что такое ассоциация?
 - б. Что такое кратность ассоциации? На какие типы подразделяются ассоциации в связи с понятием кратности?
 - в. Что такое рефлексивная ассоциация?

2. Агрегация:

а. Что такое агрегация (отношение между классами)?

б. Что такое композиция?

3. Что такое зависимость?

4. Наследственная иерархия:

а. Что такое наследование?

б. Что такое виртуальная и чисто виртуальная функция?

в. Какие классы называют конкретными, абстрактными?

г. Какие классы называют корневыми, листовыми?

д. Для чего вводится защищенная часть класса?

5. Наследование и типизация:

а. Что такое открытое (public) наследование?

б. Что такое закрытое (private) наследование?

в. Что такое защищенное (protected) наследование?

г. Допустимо ли присваивание объекту класса-родителя значения класса-потомка?

Если да, то каким образом оно реализуется?

д. Допустимо ли присваивание объекту класса-потомка значения класса-родителя?

Если да, то каким образом оно реализуется?

е. Что такое чистый полиморфизм?

ж. Что такое повышающее приведение?

з. Что такое понижающее приведение?

6. Множественное наследование:

а. Что такое множественное наследование?

б. Что такое конфликт имен?

в. Что такое повторное наследование?

г. Что такое ромбовидная структура наследования?

д. Что такое виртуальное наследование?

Лабораторная работа № 5.

Задание 1. Ознакомиться с понятиями шаблона, инстанцирования.

Задание 2. Путем модификации программ, разработанных в лабораторных работах № 3, 4, разработать шаблон контейнера для хранения объектов классов, реализующих геометрические фигуры.

2.1. Преобразовать класс-контейнер, разработанный в лабораторной работе №3, в шаблон, так чтобы элементами контейнеров могли быть различные классы, разработанные в лабораторной работе №4 (при различном инстанцировании шаблона).

2.2. Разработать функцию, демонстрирующую поведение разработанного шаблона: провести инстанцирование шаблона для каждого из классов-фигур, продемонстрировать их функционирование.

Задание 3. Подготовить текстовые файлы с разработанной программой, оттранслировать их, собрать и выполнить программу с учетом требований операционных систем и программных оболочек, в которых эта программа выполняется. При необходимости исправить ошибки и вновь повторить технологический процесс решения задачи.

Задание 4. Оформить отчет по лабораторной работе. Отчет должен содержать постановку задачи, описание разработанных классов, алгоритм, текст разработанной программы и результаты тестирования.

Задание 5. Защитить лабораторную работу, ответив на вопросы преподавателя.

Контрольные вопросы.

1. Что такое шаблон класса, как его описать на C++?

2. Что такое инстанцирование, как его описать на C++?

3. Что такое шаблон функции, как его описать на C++?

4. Каковы свойства статической переменной в языке C++?

5. Каково назначение статической функции в языке C++?
6. Что такое утилита, как ее реализовать на C++?
7. Что такое интерфейс? Как его можно описать на C++?
8. Какие отношения могут существовать между интерфейсами, между интерфейсом и классом?
9. Что такое категория классов?
10. Что такое пространство имен в языке C++?

2. Критерии оценки

- Лабораторная работа считается **не выполненной**, если студент выполнил задания частично, отчет оформил с недостатками, показал недостаточное владение практическими навыками, ответил на контрольные вопросы слишком неполно, не показал владение теоретическими навыками, оценка составляет 0-6 баллов
- Лабораторная работа считается выполненной **на пороговом** уровне, если студент выполнил задания не полностью, отчет оформил с недостатками, показал некоторое владение практическими навыками, лишь частично ответил на контрольные вопросы, показал некоторое владение теоретическими навыками, оценка составляет 7-8 баллов
- Лабораторная работа считается выполненной **на базовом** уровне, если студент выполнил задания полностью, отвечая на вопросы, не дал развернутых ответов, показал неполное владение основными концепциями и понятиями объектно-ориентированного подхода, отчет оформил без недостатков, оценка составляет 9-10 баллов
- Лабораторная работа считается выполненной **на продвинутом** уровне, если студент выполнил задания полностью, отчет оформил без недостатков, показал свободное владение основными концепциями и понятиями объектно-ориентированного подхода, оценка составляет 11-12 баллов

3. Шкала оценки

В общей оценке по дисциплине баллы за лабораторные работы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Перечни вариантов

Лабораторная работа № 1.

1. Окружность.
2. Выпуклый четырехугольник.
3. Прямоугольник.
4. Квадрат.
5. Параллелограмм.
6. Трапеция.
7. Треугольник.
8. Ромб.

Лабораторная работа № 3.

1. Пусть каждому объекту из класса, реализующего геометрическую фигуру, поставлен в соответствие некоторый числовой идентификатор. Разработать класс, реализующий понятие таблицы, в качестве ключа использовать указанный идентификатор. Функция поиска ищет элемент по его идентификатору. Таблицу реализовать на основе:

а) двунаправленного списка (обеспечить «распечатку» содержимого как в прямом, так и в обратном направлении);

- б) хеширования с цепочками;
- в) дерева двоичного поиска.

2. Пусть все объекты класса, реализующего геометрическую фигуру, разделены на несколько категорий по значению некоторого признака (ввести в рассмотрение некоторый «нетривиальный» признак либо считать отнесение фигуры к конкретной категории прерогативой пользователя). Разработать класс, реализующий понятие таблицы с дубликатами (повторяющимися ключами), в качестве ключа использовать номер категории. Функция поиска ищет элементы определенной категории. Таблицу реализовать на основе:

а) двунаправленного списка (обеспечить «распечатку» содержимого как в прямом, так и в обратном направлении);

б) хеширования с цепочками, хеш-значением является номер категории;

в) дерева двоичного поиска.

3. Разработать класс, реализующий понятие множества (с неповторяющимися элементами). Функция поиска ищет фигуры с заданными геометрическими и/или графическими характеристиками. Множество реализовать на основе:

а) двунаправленного списка (обеспечить «распечатку» содержимого как в прямом, так и в обратном направлении);

б) хеширования с цепочками, хеш-функцию построить на основе ключа поиска.

4. Разработать класс, реализующий понятие множества с дубликатами (повторяющимися элементами). Функция поиска ищет фигуры с заданными геометрическими и/или графическими характеристиками. Множество реализовать на основе:

а) двунаправленного списка (обеспечить «распечатку» содержимого как в прямом, так и в обратном направлении);

б) хеширования с цепочками, хеш-функцию построить на основе ключа поиска.

5. На основе списка разработать класс, реализующий понятие:

а) «открытого» стека;

б) «открытой» очереди;

в) «открытого» дека.

«Открытость» контейнера подразумевает возможность просматривать элементы в контейнере, осуществлять поиск. Функция поиска ищет фигуры с заданными геометрическими и/или графическими характеристиками.

6. Разработать класс, реализующий понятие таблицы (см. задание к варианту 1):

а) неупорядоченной таблицы;

б) упорядоченной таблицы;

в) хеш-таблицы.

При реализации использовать массив, память под который выделяется и освобождается динамически, максимальный размер таблицы задается как параметр конструктора. Объекты класса «фигура» хранятся в контейнере по ссылке (указателю).

7. Разработать класс, реализующий понятие таблицы с дубликатами (см. задание к варианту 2):

а) неупорядоченной таблицы;

б) упорядоченной таблицы;

в) хеш-таблицы.

При реализации использовать массив, память под который выделяется и освобождается динамически, максимальный размер таблицы задается как параметр конструктора. Объекты класса «фигура» хранятся в контейнере по ссылке (указателю).

8. Разработать класс, реализующий понятие

а) «открытого» стека;

б) «открытой» очереди;

в) «открытого» дека.

Контейнер реализуется на основе массива, память под который выделяется и освобождается динамически, размер массива (максимальная вместимость контейнера) задается как параметр конструктора. «Открытость» контейнера подразумевает возможность просматривать элементы в контейнере, осуществлять поиск (функция поиска ищет фигуры с заданными геометрическими и/или графическими характеристиками). Объекты класса «фигура» хранятся в контейнере по ссылке (указателю).

Правила аттестации студентов по учебной дисциплине Объектно-ориентированное программирование

Текущая аттестация студента проводится по результатам выполнения им лабораторных работ и РГЗ, а также по результатам работы на лекциях.

Баллы за каждую из пяти лабораторных работ и РГЗ выставляются по правилам, приведенным в таблице 1.

Таблица 1

Вид деятельности	Минимальное количество баллов	Максимальное количество баллов
Выполнение лабораторной работы, РГЗ (оценивается полнота выполнения задания, включая полноту использования изученных концепций и понятий объектно-ориентированного подхода, работоспособность разработанного программного обеспечения, полнота набора тестов, полнота отчета, его соответствие требованиям оформления, своевременность* выполнения лабораторной работы, РГЗ)	4	6
Защита (оценивается полнота и глубина теоретических знаний, умение применить знания на практике, своевременность* защиты)	3	6
Всего за лабораторную работу, РГЗ	7	12

* график выполнения и защиты лабораторных работ, РГЗ приведен ниже

Минимальное допустимое количество баллов за лабораторную работу, РГЗ получает студент, выполнивший задание не полностью, не показавший владение основными концепциями и понятиями объектно-ориентированного подхода, оформивший отчет с недостатками и показавший на защите некоторое владение теоретическими и практическими навыками. Максимальное допустимое количество баллов за лабораторную работу получает студент, выполнивший задание полностью и своевременно, показавший свободное владение изученными концепциями и понятиями объектно-ориентированного подхода, своевременно оформивший отчет и своевременно защитивший лабораторную работу, показавший на защите свободное владение теоретическими и практическими навыками.

Студент, не выполнивший и не защитивший лабораторную работу, не допускается к выполнению следующей лабораторной работы.

График выполнения и защиты лабораторных работ, РГЗ:

- 1 – 4 неделя – 1-я лабораторная работа;
- 5 – 7 неделя – 2-я лабораторная работа;
- 8 – 11 неделя – 3-я лабораторная работа;
- 12 – 14 неделя – 4-я лабораторная работа;
- 15 – 18 неделя – 5-я лабораторная работа, РГЗ.

За работу на лекциях студент получает от 0 до 8 баллов (по 1 баллу за каждую лекцию).

Оценка за контрольные недели выставляется в соответствии с таблицей 2.

Таблица 2

Номер недели	7 контрольная неделя			12 контрольная неделя		
Оценка за контрольную неделю	0	1	2	0	1	2
Текущий рейтинг студента	Менее 7	От 7 до 13	14 и более	Менее 18	От 18 до 32	21 и более
Количество зачтенных лабораторных заданий	0	1	2	Менее 3	3*	3

* засчитывается выполнение лабораторной работы без защиты

Студент, выполнивший и защитивший все лабораторные работы и РГЗ и набравший не менее 42 баллов, допускается к итоговой аттестации – сдаче теоретического зачета. Студент сдает теоретический зачет в письменной форме в виде решения комплексного задания. Максимальное количество баллов за теоретический зачет – 20 баллов, минимальное – 8. Минимальное количество баллов за теоретический зачет получает студент, частично выполнивший часть отдельных заданий, составляющих комплексное задание. Максимальное количество баллов за теоретический зачет получает студент, полностью выполнивший все задания, составляющие комплексное задание.

Итоговая оценка в соответствии с 15-уровневой шкалой оценок ECTS и в традиционной форме («зачтено») выставляется на основе набранных баллов (таблица 3).

Таблица 3

Характеристика работы студента	Диапазон баллов рейтинга	Оценка ECTS	Традиционная (4-уровневая) шкала оценки
«Отлично» – работа высокого качества, уровень выполнения отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	98-100	A+	Зачтено
	94-97	A	
	90-93	A-	
«Очень хорошо» – работа хорошая, уровень выполнения отвечает большинству требований, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом в основном сформированы, все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному	87-89	B+	
	83-86	B	
	80-82	B-	

«Хорошо» – уровень выполнения работы отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки	77-79	C+	
	73-76	C	
	70-72	C-	
«Удовлетворительно» – уровень выполнения работы отвечает большинству основным требованиям, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнены, некоторые виды заданий выполнены с ошибками	67-69	D+	
	63-66	D	
	60-62	D-	
«Посредственно» – работа слабая, уровень выполнения не отвечает большинству требований, теоретическое содержание курса освоено частично, некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены, либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному	50-59	E	
«Неудовлетворительно» (с возможностью пересдачи) – теоретическое содержание курса освоено частично, необходимые практические навыки работы не сформированы, большинство предусмотренных программой обучения учебных заданий не выполнено, либо качество их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий	25-49	FX	Не зачтено
«Неудовлетворительно» (без возможности пересдачи) – теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, все выполненные учебные задания содержат грубые ошибки; дополнительная самостоятельная работа над материалом курса не приведет к какому-либо значительному повышению качества выполнения учебных заданий	0-24	F	