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

Кафедра автоматизированных систем управления

“УТВЕРЖДАЮ”
ДИРЕКТОР ИСТР
д.соц.н. Осьмук Л. А.
“___”_____ _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Программирование

Образовательная программа: 09.03.01 Информатика и вычислительная техника, профиль: Автоматизированные системы обработки информации и управления в социальной сфере

Курс: 2, семестр : 3

Институт социальных технологий и реабилитации,

		Семестр
№	**Вид деятельности**	**3**
1	Всего зачетных единиц (кредитов)	4
2	Всего часов	144
3	Всего занятий в контактной форме, час.	85
4	Лекции, час.	36
5	Практические занятия, час.	0
6	Лабораторные занятия, час.	36
7	из них в активной и интерактивной форме, час.	36
8	Аттестация, час.	2
9	Консультации, час.	11
10	Самостоятельная работа, час.	59
11	Виды самостоятельной работы (курсовой проект, курсовая работа, РГЗ, подготовка к контрольной работе)	КР
12	Вид аттестации	Э

Новосибирск 2017

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 09.03.01 Информатика и вычислительная техника

ФГОС введен в действие приказом №5 от 12.01.2016 г. , дата утверждения: 09.02.2016 г.

Место дисциплины в структуре учебного плана: Блок 1, базовая

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 09.03.01 Информатика и вычислительная техника

Рабочая программа обсуждена на заседании кафедры

АСУ, протокол заседания кафедры №7 от 20.06.2017

Утверждена на совете института социальных технологий и реабилитации, протокол № 6 от 21.06.2017

Программу разработал:

доцент, к.т.н. Достовалов Д. Н.

Заведующий кафедрой:

профессор, д.т.н. Гриф М. Г.

Ответственный за образовательную программу:

заведующий кафедрой Гриф М. Г.

1. Внешние требования

Таблица 1.1

***Компетенция ФГОС*: ОПК.5 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; *в части следующих результатов обучения*:**
з4. знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах
у12. уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе
у5. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств
***Компетенция ФГОС*: ПК.3 способность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности; *в части следующих результатов обучения*:**
у7. уметь осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности
***Компетенция НГТУ*: ПК.9.В/ПК готовность к разработке моделей компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина"; *в части следующих результатов обучения*:**
з2. знать основы объектно-ориентированного подхода к программированию

2. Требования НГТУ к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий
Программирование	
ПК.9.В/ПК.з2 знать основы объектно-ориентированного подхода к программированию	
1.знать основы объектно-ориентированного подхода к программированию	Лекции; Самостоятельная работа
ПК.3.у7 уметь осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	
2.уметь осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	Лабораторные работы; Самостоятельная работа
ОПК.5.з4 знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах	
3.знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах	Лекции; Самостоятельная работа
ОПК.5.у12 уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе	
4.уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе	Лабораторные работы; Самостоятельная работа
ОПК.5.у5 уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	
5.уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	Лабораторные работы; Самостоятельная работа

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лекций	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 3				
Дидактическая единица: Введение в ООП				
1. Два фундаментальных подхода к разработке программ - структурное программирование и объектно-ориентированное программирование.	0	1	1, 3	Подготовка к лекции, посещение лекции, чтение литературы
2. Введение. Структура курса и правила аттестации.	0	1	1, 3	Подготовка к лекции, посещение лекции, чтение литературы
Дидактическая единица: Парадигма ООП				
3. Парадигма ООП. Основные понятия ООП - класс и объект. Три базовых принципа ООП - инкапсуляция, наследование и полиморфизм.	2	2	1, 3	Подготовка к лекции, посещение лекции, чтение литературы
Дидактическая единица: Инкапсуляция				
4. Инкапсуляция. Определение класса. Реализация абстрактного типа данных в форме класса. Область действия класса и доступ к его элементам. Отделение интерфейса класса от реализации класса. Управление доступом к элементам класса. Функции доступа и функции-утилиты. Инициализация объектов класса. Конструкторы. Конструкторы без параметров и с параметрами. Деструкторы. Порядок вызова конструкторов и деструкторов. Присваивание объектов - поэлементное копирование объектов по умолчанию. Константные объекты, данные-элементы и функции-элементы классов. Композиция (объекты классов как элементы других классов). Дружественные функции. Дружественные классы. Указатель this. Динамическое выделение памяти посредством операторов new и delete. Конструкторы копий. Статические элементы класса. Контейнерные классы и итераторы.	2	6	1, 3	Подготовка к лекции, посещение лекции, чтение литературы
Дидактическая единица: Наследование				

5. Наследование. Базовые и производные классы. Защищенные элементы класса. Преобразование указателей базового класса в указатели производного класса. Переопределение функций-элементов базового класса в производном классе. Открытое, защищенное и закрытое наследование. Конструкторы и деструкторы в производных классах. Неявное преобразование объектов производного класса в объекты базового класса. Сопоставление композиции и наследования. Пример иерархии классов. Прямое и косвенное наследование. Множественное наследование. Виртуальные базовые классы.	2	6	1, 3	Подготовка к лекции, посещение лекции, чтение литературы
Дидактическая единица: Полиморфизм				
6. Полиморфизм. Виртуальные функции. Абстрактные и конкретные классы. Динамическое связывание как средство реализации динамического полиморфизма. Виртуальные деструкторы.	2	4	1, 3	Подготовка к лекции, посещение лекции, чтение литературы
Дидактическая единица: Перегрузка операций				
7. Перегрузка операций. Принципы перегрузки операций. Ограничения на перегрузку операций. Функции-операции как элементы и как друзья класса. Перегрузка операций включения в поток и извлечения из потока. Перегрузка унарных операций. Перегрузка бинарных операций. Преобразование типов. Перегрузка операций инкремента и декремента.	2	4	1, 3	Подготовка к лекции, посещение лекции, чтение литературы
Дидактическая единица: Потоки ввода-вывода				
8. Потоки ввода-вывода. Потоки. Заголовочные файлы библиотеки iostream. Классы и объекты потокового ввода-вывода. Потоковый вывод. Операция включения в поток. Потоковый ввод. Операция извлечения из потока. Функции бесформатного ввода-вывода. Манипуляторы потока. Состояние формата потока. Состояния ошибки потоков. Привязка потока вывода к потоку ввода.	2	4	1, 3	Подготовка к лекции, посещение лекции, чтение литературы
Дидактическая единица: Шаблоны функций и классов				

9. Шаблоны функций и классов. Друзья классов в сочетании с шаблонами классов. Наследование в сочетании с шаблонами классов. Статические элементы классов и шаблоны классов.	2	4	1, 3	Подготовка к лекции, посещение лекции, чтение литературы
Дидактическая единица: Обработка исключений				
10. Обработка исключений. Условия применения обработки исключений. Средства управления исключениями: оператор выброса исключения (throw), пробный блок (try), блок перехвата (catch). Выброс исключения. Перехват исключения. Перебрасывание исключения. Спецификация исключений. Обработка непредусмотренных исключений. Разматывание стека. Конструкторы и деструкторы в управлении исключениями. Исключения при наследовании.	2	2	1, 3	Подготовка к лекции, посещение лекции, чтение литературы
Дидактическая единица: Разработка промышленных программных продуктов на основе технологии ООП				
11. Создание, тестирование и отладка сложных программных продуктов на базе технологии ООП в коммерческих инструментальных средах визуальной разработки программ Borland Developer Studio и Microsoft Visual Studio .NET	2	2	1, 3	Подготовка к лекции, посещение лекции, чтение литературы

Таблица 3.2

Темы лабораторных работ	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 3				
Дидактическая единица: Парадигма ООП				
1. Создание полнофункционального приложения Windows с графическим интерфейсом пользователя в инструментальной среде C++Builder.	2	4	2, 4, 5	Студенты создают приложение "Текстовый редактор" с экранными окнами, строкой меню, панелью инструментов и строкой состояния.
Дидактическая единица: Инкапсуляция				
2. Создание приложения Windows с графическим интерфейсом пользователя в инструментальной среде Microsoft Visual Studio.	2	4	2, 4, 5	Студенты создают приложение "Текстовый редактор" с экранными окнами, строкой меню, панелью инструментов и строкой состояния.

3. Создание приложений Windows с графическим интерфейсом в среде программирования Qt.	2	4	2, 4, 5	Студенты создают приложение "Текстовый редактор" с экранными окнами, строкой меню, панелью инструментов и строкой состояния.
4. Знакомство с языками программирования Java и C#. Создание приложений с графическим интерфейсом в среде Intellij Idea и MS Visual Studio.	2	4	2, 4, 5	Студенты создают приложение с графическим интерфейсом пользователя, изучают способы компоновки экранных форм и основные элементы управления.
5. Применение объектно-ориентированного подхода к анализу, проектированию и программированию. Изучение принципов инкапсуляции и агрегации (композиции).	2	4	2, 4, 5	Студенты создают приложение, которое использует класс, представляющий заданную предметную область (например, сеть магазинов, система матриц, игра в шашки и др.) и предоставляющий удобный графический интерфейс пользователю (менеджер, математик, игрок и т.п.).
Дидактическая единица: Наследование				
6. Применение объектно-ориентированного подхода. Изучение принципа наследования.	2	4	2, 4, 5	Студенты дорабатывают ранее созданное приложение, вводя иерархию классов (наследование).
Дидактическая единица: Полиморфизм				
7. Применение объектно-ориентированного подхода. Изучение и использование полиморфизма.	2	4	2, 4, 5	Студенты дорабатывают ранее созданное приложение, определяя и вызывая виртуальный функции. Используется статическое и динамическое связывание.
Дидактическая единица: Потоки ввода-вывода				
8. Применение объектно-ориентированного подхода. Изучение объектов потокового ввода и вывода, а также способов перегрузки операций для работы с потоками.	2	4	2, 4, 5	Студенты дорабатывают ранее созданное приложение: определяют для ранее созданных классов операции обмена с потоком, создают и используют объекты файловых потоков.
Дидактическая единица: Шаблоны функций и классов				
9. Применение объектно-ориентированного подхода и обобщенного программирования. Изучение динамических структур данных. Разработка шаблонных методов и классов.	2	4	2, 4, 5	Студенты дорабатывают ранее созданное приложение. Создают шаблон динамической структуры данных (список, стек, очередь и др.). Помещают в структуру данных объекты ранее разработанных классов.

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации

Семестр: 3				
1	Курсовая работа	1, 2, 3, 4, 5	32	4
: Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана.				
2	Подготовка к занятиям	1, 2, 3, 4, 5	20	6
: Программирование : методические указания к лабораторным работам для 2 курса АВТФ направления 230100 "Информатика и вычислительная техника" дневного отделения / Новосиб. гос. техн. ун-т ; [сост. Р. Г. Шахмаметов]. - Новосибирск, 2013. - 18, [2] с.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000179535 Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана. Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс [для студентов АВТФ по направлению 230100 Инфоматика и вычислительная техника] / Р. Г. Шахмаметов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2013]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000183229. - Загл. с экрана.				
3	Подготовка к аттестации	1, 2, 3, 4, 5	7	1
: Программирование : методические указания к лабораторным работам для 2 курса АВТФ направления 230100 "Информатика и вычислительная техника" дневного отделения / Новосиб. гос. техн. ун-т ; [сост. Р. Г. Шахмаметов]. - Новосибирск, 2013. - 18, [2] с.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000179535 Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана. Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс [для студентов АВТФ по направлению 230100 Инфоматика и вычислительная техника] / Р. Г. Шахмаметов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2013]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000183229. - 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	e-mail:d.dostovalov@corp.nstu.ru; Портал НГТУ:Система отправки сообщений студентам; Среда электронного обучения НГТУ:http://dispace.edu.nstu.ru/didesk/course/show/5420
Консультирование	e-mail: d.dostovalov@corp.nstu.ru; Среда электронного обучения НГТУ:http://dispace.edu.nstu.ru/didesk/course/show/5420
Размещение учебных материалов	Среда электронного обучения НГТУ:http://dispace.edu.nstu.ru/didesk/course/show/5420

Таблица 5.2

Активные и интерактивные формы проведения занятий

№	Наименование активных форм	Коды формируемых компетенций
1	Дискуссия	ОПК.5; ПК.9.В/ПК
Формируемые умения: з2. знать основы объектно-ориентированного подхода к программированию; з4. знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах; у12. уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе; у5. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств		
Краткое описание применения: Обсуждение теоретических положений, сопоставление полученных результатов с ожидаемыми, ответы на контрольные вопросы		

6. Правила аттестации обучающихся по учебной дисциплине

Для аттестации обучающихся по дисциплине используется балльно-рейтинговая система (БРС), позволяющая выставлять оценки по традиционной шкале и 15-уровневой ECTS. Краткая информация о БРС приведена в табл. 6.1.

Таблица 6.1

Оцениваемые виды деятельности обучающихся	Мин. балл	Максимальный балл
Семестр: 3		
Лабораторная №1: Создание приложения Windows в инструментальной среде C++Builder	2	4
Контролирующие материалы приводятся в "Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана."		
Лабораторная №2: Создание приложения Windows в инструментальной среде Microsoft Visual Studio	2	4
Контролирующие материалы приводятся в "Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана."		
Лабораторная №3: Создание приложений Windows в среде программирования Qt	2	4
Контролирующие материалы приводятся в "Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана."		
Лабораторная №4: Знакомство с языками программирования Java и C#. Создание приложений с графическим интерфейсом в среде Intellij Idea и MS Visual Studio	2	4
Контролирующие материалы приводятся в "Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана."		
Лабораторная №5: Применение объектно-ориентированного подхода к анализу, проектированию и программированию. Изучение принципов инкапсуляции и агрегации (композиции)	2	10
Контролирующие материалы приводятся в "Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана."		

Лабораторная №6: Применение объектно-ориентированного подхода. Изучение принципа наследования	2	10
Контролирующие материалы приводятся в "Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана."		
Лабораторная №7: Применение объектно-ориентированного подхода. Изучение и использование полиморфизма	2	8
Контролирующие материалы приводятся в "Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана."		
Лабораторная №8: Применение объектно-ориентированного подхода. Изучение объектов потокового ввода и вывода, а также способов перегрузки операций для работы с потоками	2	8
Контролирующие материалы приводятся в "Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана."		
Лабораторная №9: Применение объектно-ориентированного подхода и обобщенного программирования. Изучение динамических структур данных. Разработка шаблонных методов и классов	2	8
Контролирующие материалы приводятся в "Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана."		
Курсовая работа:	0	100
Контролирующие материалы приводятся в "Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана."		
Экзамен:	0	40
Контролирующие материалы приводятся в "Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - 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 Л/Р	**Защита КП/КР**	**Экзамен**
ОПК.5	з4. знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах	+	+	+
	у12. уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе	+	+	+
	у5. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	+	+	+
ПК.3	у7. уметь осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	+	+	+
	ПК.9.В/ПК з2. знать основы объектно-ориентированного подхода к программированию	+	+	+

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

7. Литература

Основная литература

1. Архангельский А. Я. Программирование в C++Builder 6 и 2006 / А. Я. Архангельский, М. А. Тагин. - М., 2007. - 1181 с. : ил. + 1 CD-ROM.

2. Биллиг В. А. Основы объектного программирования на C# : (C# 3.0, Visual Studio 2008) : учебное пособие / В. А. Биллиг. - М., 2010. - 582 , [1] с. : ил., табл.

3. Кнут Д. Э. Искусство программирования. Т. 1 : учебное пособие / Дональд Э. Кнут. - М., 2011

4. Шилдт Г. C++. Базовый курс : [полное описание языка C++ от элементарных понятий до супервозможностей, множество советов, рекомендаций и сотни примеров, информация о стандартной библиотеке шаблонов (STL)] / Герберт Шилдт ; [пер. с англ. и ред. Н. М. Ручко]. - М. [и др.], 2011. - 620 с. : ил., табл.. - На обл.: Изучайте C++ с профессионалом!.

5. Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс / Р. Г. Шахмаметов, Д. Н. Достовалов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000222049. - Загл. с экрана.

6. Романов Е. Л. Объектно-ориентированное программирование [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156348. - Загл. с экрана.

7. Культин Н. Б. Microsoft Visual C++ в задачах и примерах : [базовые компоненты, программирование, графики и баз данных, справочник по компонентам и функциям] / Никита Культин. - СПб., 2011. - 264 с. : ил., табл. + 1 CD-ROM.

8. Березин Б. И. Начальный курс C и C++ : [учебное пособие] / Б. И. Березин, С. Б. Березин. - М., 2012. - 280 с.

9. Дейтел Х. М. Как программировать на C++ / Х. М. Дейтел, П. Дж. Дейтел ; пер. с англ. под ред. В. В. Тимофеева. - М., 2007. - 799 с. : ил.

10. Культин Н. Б. C++ Builder / Никита Культин. - СПб., 2008. - 463 с. : ил., табл. + 1 CD-ROM.

11. Культин Н. Б. Основы программирования Microsoft Visual C++ 2010 / Никита Культин. - СПб., 2010. - 376 с. : ил., табл. + 1 CD-ROM.

12. Лаптев В. В. C++. Объектно-ориентированное программирование : [учебное пособие для вузов] / В. В. Лаптев. - СПб. [и др.], 2008. - 457 с. : ил.. - Издательская программа "300 лучших учебников для высшей школы".

13. Лафоре Р. Объектно-ориентированное программирование в C++ / Р. Лафоре ; [пер. с англ. А. Кузнецова, М. Назарова, В. Шрага]. - СПб., 2007. - 923 с. : ил.

14. Павловская Т. А. C#. Программирование на языке высокого уровня : [учебник по направлению "Информатика и вычислительная техника"] / Т. А. Павловская. - СПб. [и др.], 2010. - 432 с. : ил.

15. Павловская Т. А. C/C++. Программирование на языке высокого уровня : [учебник по направлению "Информатика и вычислительная техника"] / Т. А. Павловская. - СПб. [и др.], 2010. - 460 с. : табл., ил.

16. Прата С. Язык программирования C++ : лекции и упражнения / Стивен Прата ; [пер. с англ. Д. Я. Иваненко и др.]. - М. [и др.], 2007. - 1181 с. : ил.

17. Приемы объектно-ориентированного проектирования. Паттерны проектирования : [пер. с англ.] / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. - СПб. [и др.], 2008. - 366 с. : ил.

18. Рамбо Д. UML 2.0. Объектно-ориентированное моделирование и разработка / Дж. Рамбо, М. Блаха. - СПб. [и др.], 2007. - 540, [4] с. : ил.

19. Солтер Н. А. C++ для профессионалов / Николас А. Солтер, Скотт Дж. Клепер. - М. [и др.], 2006. - 904 с. : ил.

20. Стэкер М. А. Разработка клиентских Windows-приложений на платформе Microsoft .NET Framework : экзамен 70-526 MCTS : [пер. англ.] / Мэтью А. Стэкер, Стивен Дж. Стэйн, Тони Нортроп. - М. [и др.], 2008. - XX, 602 с. : ил. + 1 CD-ROM.

21. Троелсен Э. C# и платформа .NET / Эндрю Троелсен. - СПб. [и др.], 2007. - 795 с. : ил.

22. Троелсен Э. Язык программирования C#2010 и платформа. NET 4.0 / Э. Троелсен. - М., 2011

23. Хорев П. Б. Объектно-ориентированное программирование : [учебное пособие по направлению "Информатика и вычислительная техника"] / П. Б. Хорев. - М., 2011. - 446, [1] с. : ил.

24. Шамис В. А. C++ Builder Borland Developer Studio 2006 / В. А. Шамис. - СПб. [и др.], 2007. - 780 с. : ил.

25. Шеферд Д. Программирование на Microsoft Visual C++ .NET : мастер-класс [пер. с англ.] / Джордж Шеферд по материалам Дэвида Круглински. - М., 2007. - 892 с. : ил. + 1 CD-ROM.

26. Шилдт Г. C++ : для начинающих : самоучитель / Герберт Шилдт ; пер. с англ. К. Г. Финогенова. - М., 2007. - 639 с. : ил.

27. Шилдт Г. Самоучитель C++ : пер. с англ. / Герберт Шилдт. - СПб., 2007. - 683 с. + 1 CD-ROM.

28. Шахмаметов Р. Г. Программирование [Электронный ресурс] : электронный учебно-методический комплекс [для студентов АВТФ по направлению 230100 Инфоматика и вычислительная техника] / Р. Г. Шахмаметов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2013]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000183229. - Загл. с экрана.

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.

2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Гради Буч ; пер. с англ. под ред. И. Романовского и Ф. Андреева. - М., 1998. - 558 с. : ил.. - Тит. л. парал. рус., англ..

3. Кубенский А. А. Структуры и алгоритмы обработки данных: объектно-ориентированный подход и реализация на C++ : учебное пособие по специальности "Математическое обеспечение и администрирование информационных систем" - 351500 / А. А. Кубенский. - Санкт-Петербург, 2004. - 464 с. : ил. + 1 CD-ROM.

4. Страуструп Б. Язык программирования C++. Специальное издание / Б. Страуструп ; пер. с англ. С. Анисимова и М. Кононова ; под. ред. Ф. Андреева и А. Ушакова. - М., 2005. - 1096 с. : ил.. - С авт. изм. и доп..

5. Франка П. Б. C++ : учебный курс : / П. Франка ; [пер. с англ. П. Бибиков]. - М. [и др.], 2005. - 521 с. : ил.. - Тит. л. парал. рус., англ..

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 "Znanium.com" : http://znanium.com/

5. :

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

1. Культин Н. Б. C/C++ в задачах и примерах / Н. Культин. - СПб., 2006. - 281 с. : ил.

2. Лаптев В. В. C++. Объектно-ориентированное программирование. Задачи и упражнения : [учебное пособие для вузов по направлению "Информатика и вычислительная техника"] / В. В. Лаптев, А. В. Морозов, А. В. Бокова. - СПб. [и др.], 2007. - 287 с. : табл.

3. Программирование : методические указания к лабораторным работам для 2 курса АВТФ направления 230100 "Информатика и вычислительная техника" дневного отделения / Новосиб. гос. техн. ун-т ; [сост. Р. Г. Шахмаметов]. - Новосибирск, 2013. - 18, [2] с.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000179535

8.2 Специализированное программное обеспечение

1 Visual Studio 2013
2 Visual Studio 2015
3 Microsoft Visio
4 QtSDK
5 C++Builder 2007 Professional R2

9. Материально-техническое обеспечение

Презентационное оборудование

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

Компьютерный класс

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

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

Кафедра автоматизированных систем управления

“УТВЕРЖДАЮ”
ДИРЕКТОР ИСТР
д.соц.н., профессор Л.А. Осьмук
“___ ”_____ _____ г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Программирование

Образовательная программа: 09.03.01 Информатика и вычислительная техника, профиль: Автоматизированные системы обработки информации и управления в социальной сфере

Новосибирск 2017

1. **Обобщенная структура фонда оценочных средств учебной дисциплины**

Обобщенная структура фонда оценочных средств по **дисциплине** 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.5 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	з4. знать современные технические и программные средства взаимодействия с вычислительной техникой, технологию разработки алгоритмов и программ, методы отладки и решения задач на вычислительной технике в различных режимах	Введение. Структура курса и правила аттестации. Два фундаментальных подхода к разработке программ - структурное программирование и объектно-ориентированное программирование. Инкапсуляция. Определение класса. Реализация абстрактного типа данных в форме класса. Область действия класса и доступ к его элементам. Отделение интерфейса класса от реализации класса. Управление доступом к элементам класса. Функции доступа и функции-утилиты. Инициализация объектов класса. Конструкторы. Конструкторы без параметров и с параметрами. Деструкторы. Порядок вызова конструкторов и деструкторов. Присваивание объектов - поэлементное копирование объектов по умолчанию. Константные объекты, данные-элементы и функции-элементы классов. Композиция (объекты классов как элементы других классов). Дружественные функции. Дружественные классы. Указатель this. Динамическое выделение памяти посредством операторов new и delete. Конструкторы копий. Статические элементы класса. Контейнерные классы и итераторы. Наследование. Базовые и производные классы. Защищенные элементы класса. Преобразование указателей базового класса в указатели производного класса. Переопределение функций-элементов базового класса в производном классе. Открытое, защищенное и закрытое наследование. Конструкторы и деструкторы	Курсовая работа, Отчеты по лабораторным работам №№ 1–9.	Экзамен, вопросы №№ 1–34.

		в производных классах. Неявное преобразование объектов производного класса в объекты базового класса. Сопоставление композиции и наследования. Пример иерархии классов. Прямое и косвенное наследование. Множественное наследование. Виртуальные базовые классы. Обработка исключений. Условия применения обработки исключений. Средства управления исключениями: оператор выброса исключения (throw), пробный блок (try), блок перехвата (catch). Выброс исключения. Перехват исключения. Перебрасывание исключения. Спецификация исключений. Обработка непредусмотренных исключений. Разматывание стека. Конструкторы и деструкторы в управлении исключениями. Исключения при наследовании. Парадигма ООП. Основные понятия ООП - класс и объект. Три базовых принципа ООП - инкапсуляция, наследование и полиморфизм. Перегрузка операций. Принципы перегрузки операций. Ограничения на перегрузку операций. Функции-операции как элементы и как друзья класса. Перегрузка операций включения в поток и извлечения из потока. Перегрузка унарных операций. Перегрузка бинарных операций. Преобразование типов. Перегрузка операций инкремента и декремента. Полиморфизм. Виртуальные функции. Абстрактные и конкретные классы. Динамическое связывание как средство реализации динамического полиморфизма. Виртуальные деструкторы. Потоки ввода-вывода. Потоки. Заголовочные файлы библиотеки iostream. Классы и объекты потокового ввода-вывода. Потоковый вывод. Операция включения в поток. Потоковый ввод. Операция извлечения из потока. Функции бесформатного ввода-вывода. Манипуляторы потока. Состояние формата потока. Состояния ошибки потоков. Привязка потока вывода к		

		потоку ввода. Создание, тестирование и отладка сложных программных продуктов на базе технологии ООП в коммерческих инструментальных средах визуальной разработки программ Borland Developer Studio и Microsoft Visual Studio .NET Шаблоны функций и классов. Друзья классов в сочетании с шаблонами классов. Наследование в сочетании с шаблонами классов. Статические элементы классов и шаблоны классов.		
ОПК.5	у5. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	Знакомство с языками программирования Java и C#. Создание приложений с графическим интерфейсом в среде Intellij Idea и MS Visual Studio. Применение объектно-ориентированного подхода и обобщенного программирования. Изучение динамических структур данных. Разработка шаблонных методов и классов. Применение объектно-ориентированного подхода. Изучение и использование полиморфизма. Применение объектно-ориентированного подхода. Изучение объектов потокового ввода и вывода, а также способов перегрузки операций для работы с потоками. Применение объектно-ориентированного подхода. Изучение принципа наследования. Применение объектно-ориентированного подхода к анализу, проектированию и программированию. Изучение принципов инкапсуляции и агрегации (композиции). Создание полнофункционального приложения Windows с графическим интерфейсом пользователя в инструментальной среде C++Builder. Создание приложений Windows с графическим интерфейсом в среде программирования Qt. Создание приложения Windows с графическим интерфейсом пользователя в инструментальной среде Microsoft Visual Studio.	Курсовая работа, Отчеты по лабораторным работам №№ 1–9.	Экзамен, вопросы №№ 1–34.

ОПК.5	у12. уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе	Знакомство с языками программирования Java и C#. Создание приложений с графическим интерфейсом в среде Intellij Idea и MS Visual Studio. Применение объектно-ориентированного подхода и обобщенного программирования. Изучение динамических структур данных. Разработка шаблонных методов и классов. Применение объектно-ориентированного подхода. Изучение и использование полиморфизма. Применение объектно-ориентированного подхода. Изучение объектов потокового ввода и вывода, а также способов перегрузки операций для работы с потоками. Применение объектно-ориентированного подхода. Изучение принципа наследования. Применение объектно-ориентированного подхода к анализу, проектированию и программированию. Изучение принципов инкапсуляции и агрегации (композиции). Создание полнофункционального приложения Windows с графическим интерфейсом пользователя в инструментальной среде C++Builder. Создание приложений Windows с графическим интерфейсом в среде программирования Qt. Создание приложения Windows с графическим интерфейсом пользователя в инструментальной среде Microsoft Visual Studio.	Курсовая работа, Отчеты по лабораторным работам №№ 1–9.	Экзамен, вопросы №№ 1–34.
ПК.3/НИ готовность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	у7. уметь осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	Знакомство с языками программирования Java и C#. Создание приложений с графическим интерфейсом в среде Intellij Idea и MS Visual Studio. Применение объектно-ориентированного подхода и обобщенного программирования. Изучение динамических структур данных. Разработка шаблонных методов и классов. Применение объектно-ориентированного подхода. Изучение и использование полиморфизма. Применение объектно-ориентированного подхода. Изучение объектов потокового ввода и вывода, а также способов перегрузки	Курсовая работа, Отчеты по лабораторным работам №№ 1–9.	Экзамен, вопросы №№ 1–34.

		операций для работы с потоками. Применение объектно-ориентированного подхода. Изучение принципа наследования. Применение объектно-ориентированного подхода к анализу, проектированию и программированию. Изучение принципов инкапсуляции и агрегации (композиции). Создание полнофункционального приложения Windows с графическим интерфейсом пользователя в инструментальной среде C++Builder. Создание приложений Windows с графическим интерфейсом в среде программирования Qt. Создание приложения Windows с графическим интерфейсом пользователя в инструментальной среде Microsoft Visual Studio.		
ПК.9.В/ПК готовность к разработке моделей компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина"	з2. знать основы объектно-ориентированного подхода к программированию	Введение. Структура курса и правила аттестации. Два фундаментальных подхода к разработке программ - структурное программирование и объектно-ориентированное программирование. Инкапсуляция. Определение класса. Реализация абстрактного типа данных в форме класса. Область действия класса и доступ к его элементам. Отделение интерфейса класса от реализации класса. Управление доступом к элементам класса. Функции доступа и функции-утилиты. Инициализация объектов класса. Конструкторы. Конструкторы без параметров и с параметрами. Деструкторы. Порядок вызова конструкторов и деструкторов. Присваивание объектов - поэлементное копирование объектов по умолчанию. Константные объекты, данные-элементы и функции-элементы классов. Композиция (объекты классов как элементы других классов). Дружественные функции. Дружественные классы. Указатель this. Динамическое выделение памяти посредством операторов new и delete. Конструкторы копий. Статические элементы класса. Контейнерные классы и итераторы. Наследование.	Курсовая работа, Отчеты по лабораторным работам №№ 1–9.	Экзамен, вопросы №№ 1–34.

		Базовые и производные классы. Защищенные элементы класса. Преобразование указателей базового класса в указатели производного класса. Переопределение функций-элементов базового класса в производном классе. Открытое, защищенное и закрытое наследование. Конструкторы и деструкторы в производных классах. Неявное преобразование объектов производного класса в объекты базового класса. Сопоставление композиции и наследования. Пример иерархии классов. Прямое и косвенное наследование. Множественное наследование. Виртуальные базовые классы. Обработка исключений. Условия применения обработки исключений. Средства управления исключениями: оператор выброса исключения (throw), пробный блок (try), блок перехвата (catch). Выброс исключения. Перехват исключения. Перебрасывание исключения. Спецификация исключений. Обработка непредусмотренных исключений. Разматывание стека. Конструкторы и деструкторы в управлении исключениями. Исключения при наследовании. Парадигма ООП. Основные понятия ООП - класс и объект. Три базовых принципа ООП - инкапсуляция, наследование и полиморфизм. Перегрузка операций. Принципы перегрузки операций. Ограничения на перегрузку операций. Функции-операции как элементы и как друзья класса. Перегрузка операций включения в поток и извлечения из потока. Перегрузка унарных операций. Перегрузка бинарных операций. Преобразование типов. Перегрузка операций инкремента и декремента. Полиморфизм. Виртуальные функции. Абстрактные и конкретные классы. Динамическое связывание как средство реализации динамического полиморфизма. Виртуальные деструкторы. Потоки ввода-вывода. Потоки. Заголовочные		

		файлы библиотеки iostream. Классы и объекты потокового ввода-вывода. Потоковый вывод. Операция включения в поток. Потоковый ввод. Операция извлечения из потока. Функции бесформатного ввода-вывода. Манипуляторы потока. Состояние формата потока. Состояния ошибки потоков. Привязка потока вывода к потоку ввода. Создание, тестирование и отладка сложных программных продуктов на базе технологии ООП в коммерческих инструментальных средах визуальной разработки программ Borland Developer Studio и Microsoft Visual Studio .NET Шаблоны функций и классов. Друзья классов в сочетании с шаблонами классов. Наследование в сочетании с шаблонами классов. Статические элементы классов и шаблоны классов.		

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по **дисциплине** проводится в 3 семестре - в форме экзамена, который направлен на оценку сформированности компетенций ОПК.5, ПК.3/НИ, ПК.9.В/ПК.

Экзамен проводится в письменной форме, по билетам. Экзаменационный билет содержит один теоретический вопрос и две задачи.

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

В 3 семестре обязательным этапом текущей аттестации является курсовая работа. Требования к выполнению курсовой работы, состав и правила оценки сформулированы в паспорте курсовой работы.

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

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.5, ПК.3/НИ, ПК.9.В/ПК, 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 автоматизированных систем управления

Паспорт экзамена

по дисциплине «Программирование», 3 семестр

1. Методика оценки

Экзамен проводится в письменной форме, по билетам. Билет формируется по следующему правилу: один теоретический вопрос выбирается из общего списка вопросов (п. 4), одна задача выбирается из числа задач №№ 1–10 , вторая задача выбирается из числа задач №№ 11–20. Полный список задач приведен в п. 5. В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ
Билет № 10
к экзамену по дисциплине «Программирование»

---

1. Переопределение функций базовых классов в производных классах.

2. Есть ли ошибки в коде?

```
Class point {
  int x, y;
  void point(int x, int y) {
    this->x = x; this.y = y;
  }
  virtual void print() = 0;
};
```

```
int main() {
  point x(2,2);
  point *y;
  point* z = new point(1,1);
}
```

3. Создайте класс *box*, представляющий коробки. Закрытые данные-элементы (ДЭ) класса должны представлять ширину, длину, высоту и цвет коробки. Включите в открытую часть класса следующие функции-элементы: конструктор с параметрами, деструктор (при необходимости), а также функцию *print*, которая печатает информацию об объекте. Добавьте в программу перегруженный оператор =, который является функцией-элементом класса *box* и возвращает результат присваивания объектов класса. Напишите функцию *main*, которая 1) создает одномерный массив объектов класса *box* размером 4 и одновременно инициализирует эти объекты; 2) меняет местами значения первого и третьего элементов массива; 3) печатает содержимое массива.

Утверждаю: зав. кафедрой ____ _____ проф. Гриф М.Г.
(подпись)
(дата)

2. **Критерии оценки**

- Ответ на экзаменационный билет считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений, при решении задачи допускает принципиальные ошибки, оценка составляет *от 0 до 10 баллов.*
- Ответ на экзаменационный билет засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений, при решении задачи допускает непринципиальные ошибки, оценка составляет *от 11 до 20 баллов.*
- Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, законы, дает характеристику процессов, явлений, проводит анализ причин, условий, может представить качественные характеристики процессов, не допускает ошибок при решении задачи, оценка составляет *от 21 до 30 баллов.*
- Ответ на экзаменационный билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы проводит сравнительный анализ подходов, проводит комплексный анализ, выявляет проблемы, предлагает механизмы решения, способен представить количественные характеристики определенных процессов, приводит конкретные примеры из практики, не допускает ошибок и способен обосновать выбор метода решения задачи, оценка составляет от 31 до 40 *баллов.*

3. Шкала оценки

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

Итоговая оценка за экзамен рассчитывается как сумма баллов, набранных за выполнение лабораторных работ и ответа на экзаменационный билет. За выполнение всех лабораторных работ студент может получить максимум 60 баллов, за экзамен – 40 баллов.

4. Вопросы к экзамену по дисциплине «Программирование»

I. Классы (инкапсуляция)

1. Объектно-ориентированный подход. Технология ООП. Основные принципы ООП.
2. Определение класса и использование классов. Отличия классов и структур.
3. Доступ к членам класса. Функции-утилиты и функции-клиенты класса. Указатель «this».
4. Отделение интерфейса класса от реализации класса.
5. Конструкторы и деструкторы. Вызов конструкторов и деструкторов.
6. Побитовое копирование объекта при присваивании. Конструктор копирования. Правило трех.
7. Передача объектов в функцию по значению, указателю и ссылке. Способы возврата объекта функцией.
8. Динамическое выделение памяти для объекта. Особенности операций new и delete. Массивы объектов. Динамические массивы объектов.
9. Константные объекты и функции-элементы.
10. Композиция.
11. Статические элементы класса.

II Наследование

12. Общие сведения о наследовании. Особенности производного класса. Базовые и производные классы.
13. Открытые, закрытые и защищенные базовые классы.
14. Переопределение функций базовых классов в производных классах.
15. Конструкторы и деструкторы при наследовании. Вызов конструкторов и деструкторов.
16. Преобразование указателей базовых классов и указателей производных классов.
17. Множественное наследование.

III Полиморфизм

18. Виртуальные функции.
19. Абстрактные базовые классы и конкретные классы.
20. Полиморфизм. Использование полиморфизма.
21. Использование полиморфизма. Статическое и динамическое связывание.

IV Перегрузка операций

22. Основы перегрузки операций. Ограничения на перегрузку операций.
23. Функции-операции как элементы класса и как друзья класса.
24. Перегрузка бинарных операций. Перегрузка арифметических операций.
25. Перегрузка бинарных операций. Перегрузка оператора присваивания.
26. Перегрузка операций сравнения и логических операций.
27. Перегрузка унарных операций.
28. Дружественные функции в качестве функций-операций.

V Шаблоны

29. Родовые функции.
30. Родовые классы.

VI Потоки ввода-вывода

31. Перегрузка операций обмена с потоком.
32. Файловые потоки ввода-вывода. Конструирование объекта потока.

VII Обработка исключений

33. Общие сведения об исключениях. Обработка исключений. Классы исключений.
34. Исключения и стек. Механизм обработки исключений.

5. Примеры задач к экзамену по дисциплине «Программирование»

Задача 1. Функция `print(time &ob)` распечатывает данные об объекте `ob`. Есть ли ошибки в следующих вызовах функции? Если есть ошибки, исправьте их.

```
time t1, t2;
t1.print(t2);
print(&t2);
print(*t2);
```

Задача 2. Как сделать базовый класс абстрактным?

Задача 3. Как исправить в классе `Time` следующее объявление:

```
void ~Time(int *ptr);
```

Задача 4. Как вызвать в производном классе функцию-элемент базового класса, если она переопределена в данном производном классе?

Задача 5. Есть ли ошибки в определении класса?

```
class Time {
  int t[3];
  Time(int h, int m, int s);
```

```
  {
    t[0]=h; t[1]=m; t[2]=s;
  }
  void print()
  {
    for(int i=1; i<3; i++)
      cout<<t[i]<<" ";
  }
};
```

Задача 6. Как обратиться к закрытым или защищенным элементам класса?

Задача 7. Есть ли ошибки в определении класса?

```
clas Time {
  int h, m, s;
public:
  Time(int, int, int);
  ~time(int h);
  void print();
};
```

Задача 8. Как вызвать деструктор?

Задача 9. Можно ли в структуру включать функции-элементы?

Задача 10. Что бы вы исправили в определении класса?

```
class Time {
  int h, m, s;
public:
  void Time(int h=0, int m=0, int s=0);
  void print();
};
```

Задача 11. Создайте класс *samp*. Включите в закрытую часть класса целочисленные переменные *a* и *b*, а в открытую часть класса – конструктор с параметрами, а также функции *geta* и *getb*, которые возвращают соответственно значения переменных *a* и *b*. Напишите функцию *print*, получающую объект класса *samp* по ссылке и печатающую значения его данных-элементов. Напишите функцию *main*, которая создает одномерный массив объектов класса *samp* размером 5 и одновременно инициализирует эти объекты значениями (1; 2), (3; 4), (5; 6), (7; 8) и (9; 10) соответственно, а затем печатает содержимое объектов из массива, начиная с конца массива.

Задача 12. Создайте класс *samp*. Включите в закрытую часть класса вещественную переменную *a* и строку *b*, а в открытую часть класса: 1) конструктор копирования, 2) функцию *init*, предназначенную для задания значений данных-элементов класса, 3) функции получения значений данных-элементов класса *geta* и *getb*, 4) деструктор (при необходимости). Напишите функцию *print*, получающую объект класса *samp* и печатающую значения его данных-элементов. Напишите функцию *main*, которая создает одномерный массив объектов класса *samp* размером 4 и одновременно инициализирует эти объекты значениями (1.1; “one”), (2.2; “two”), (3.3; “three”) и (4.4; “four”) соответственно, а затем печатает содержимое объектов из массива, начиная с начала массива и используя функцию *print*. Укажите, какие операторы будут приводить к вызову конструктора копирования при работе программы (если они есть).

Задача 13. Создайте класс *vector*, представляющий векторы в трехмерном пространстве. Включите в закрытую часть класса целочисленные переменные *x, y* и *z*, представляющие координаты вектора. Включите в открытую часть класса конструктор с параметрами, а также функции *getx, gety* и *getz*, которые возвращают соответственно значения переменных *x, y* и *z*. Напишите функцию *print*, получающую объект класса *vector* по указателю и печатающую значения его данных-элементов. Напишите функцию *main*, которая создает одномерный массив объектов класса *vector* размером 3 и одновременно

инициализирует эти объекты значениями (0; 1; 2), (2; 3; 4) и (4; 5; 6) соответственно, а затем печатает содержимое объектов из массива.

Задача 14. Создайте класс *vector*, представляющий векторы в трехмерном пространстве. Включите в закрытую часть класса целочисленные переменные *x, y* и *z*, представляющие координаты вектора. Включите в открытую часть класса конструктор с параметрами, деструктор (при необходимости), а также функцию *print*, которая печатает информацию об объекте. Добавьте в программу перегруженный оператор + , который является функцией-элементом класса *vector* и возвращает результат сложения объектов класса. Напишите функцию *main*, которая 1) создает одномерный массив объектов класса *vector* размером 4 и одновременно инициализирует эти объекты значениями (0; 1; 2), (2; 3; 4), (4; 5; 6) и (6; 7; 8) соответственно, 2) печатает результат сложения первых трех объектов из массива.

Задача 15. Создайте класс *vector*, представляющий векторы в трехмерном пространстве. Включите в закрытую часть класса целочисленные переменные *x, y* и *z*, представляющие координаты вектора, а в открытую часть класса: 1) конструктор копирования, 2) функцию *init*, предназначенную для задания значений данных-элементов класса, 3) функции получения значений данных-элементов класса *getx, gety* и *getz*, 4) деструктор (при необходимости). Напишите функцию *print*, получающую ссылку на объект класса *vector* и печатающую значения его данных-элементов. Напишите функцию *main*, которая создает одномерный массив объектов класса *vector* размером 3 и одновременно инициализирует эти объекты значениями (0; 1; 2), (2; 3; 4) и (4; 5; 6) соответственно, а затем печатает содержимое объектов из массива, начиная с начала массива и используя функцию *print*. Укажите, какие операторы будут приводить к вызову конструктора копирования при работе программы (если они есть).

Задача 16. Создайте класс *box*, представляющий коробки. Закрытые данные-элементы (ДЭ) класса должны представлять ширину, длину, высоту и цвет коробки. Включите в открытую часть класса следующие функции-элементы: конструктор с параметрами; конструктор копий, который устанавливает в копии для цвета коробки значение “зеленый”, а для остальных ДЭ класса – значения одноименных ДЭ из объекта-оригинала; функция *show*, печатающая содержимое объекта класса.
Напишите функцию *main*, которая создает объект класса *box*, представляющий коробку красного цвета с размерами 2 x 3 x 8, затем посредством конструктора копий создает копию этого объекта, представляющую коробку зеленого цвета, и в заключение печатает содержимое объекта и копии.

Задача 17. Создайте класс *box*, представляющий коробки. Закрытые данные-элементы (ДЭ) класса должны представлять ширину, длину, высоту и цвет коробки. Включите в открытую часть класса следующие функции-элементы: конструктор с параметрами, деструктор (при необходимости), а также функцию *print*, которая печатает информацию об объекте. Добавьте в программу перегруженный оператор = , который является функцией-элементом класса *box* и возвращает результат присваивания объектов класса. Напишите функцию *main*, которая 1) создает одномерный массив объектов класса *box* размером 4 и одновременно инициализирует эти объекты, 2) меняет местами значения первого и третьего элементов массива, 3) печатает содержимое массива.

Задача 18. Создайте класс *box*, представляющий коробки. Закрытые данные-элементы (ДЭ) класса должны представлять ширину, длину и высоту коробки. В открытую часть класса включите: 1) конструктор копирования, 2) функцию *init*, предназначенную для задания значений данных-элементов класса, 3) функции получения значений данных-элементов класса, 4) деструктор (при необходимости). Напишите функцию *volume*, получающую указатель на объект класса *box* и печатающую объем коробки. Напишите функцию *main*, которая создает одномерный массив объектов класса *box* размером 4 и одновременно инициализирует эти объекты значениями (0; 1; 2), (2; 3; 4), (4; 5; 6) и (7; 8; 9) соответственно, а затем печатает объем коробок из массива, начиная

с конца массива и используя функцию *volume*. Укажите, какие операторы будут приводить к вызову конструктора копирования при работе программы (если они есть).

Задача 19. Создайте иерархию классов:

Класс	Базовый класс	Абстрактный объект (АО), представляемый классом	
		Название АО	Свойства АО
`Food`	–	продукт питания	изготовитель, поставщик, цена
`Meat`	`Food`	мясное изделие	дата изготовления
`Sausage`	`Meat`	колбасное изделие	тип, вид
`Drink`	`Food`	напиток	емкость тары
`Juice`	`Drink`	сок	состав, % сахара

Сделайте данные-элементы базовых классов *защищенными*, а производных классов – *закрытыми*. Включите в каждый класс *виртуальную* функцию *show*, которая печатает содержимое объекта класса. Включите в классы дополнительные функции-элементы, если они необходимы. Напишите функцию *main*, которая создает объекты классов *Fresh* и *Juice* и печатает их содержимое. Сообщите, какой метод вызова функции *show* (статическое или динамическое связывание) вы применили, и обоснуйте свой ответ.

Задача 20. Создайте иерархию классов:

Класс	Базовый класс	Абстрактный объект (АО), представляемый классом	
		Название АО	Свойства АО
`Shape`	–	геометрическая фигура	цвет
`Shape2D`	`Shape`	плоская фигура	-
`Circle`	`Shape2D`	круг	радиус
`Square`	`Shape2D`	квадрат	длина стороны
`Shape3D`	`Shape`	объемная фигура	-
`Cube`	`Shape3D`	куб	длина стороны

Сделайте данные-элементы базовых и производных классов классов *защищенными*. Включите в каждый класс *виртуальную* функцию *volume*, которая печатает площадь или объем фигуры. Сделайте классы *Shape*, *Shape2D* и *Shape3D* абстрактными. Включите в классы дополнительные функции-элементы, если они необходимы. Напишите функцию *main*, которая создает объекты классов *Circle* и *Cube* и печатает их площадь или объем. Сообщите, какой метод вызова функции *volume* (статическое или динамическое связывание) вы применили, и обоснуйте свой ответ.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра автоматизированных систем управления

Паспорт
курсовой работы

по дисциплине «Программирование», 3 семестр

1. **Методика оценки.**

1.1 Содержание пояснительной записки

- Титульный лист.
- Оглавление.
- Постановка задачи.
- Описание классов.
- Взаимосвязь классов.
- Описание алгоритмов, относящихся к предметной области (например, алгоритм моделирования или алгоритм выбора хода в игре).
- Описание разработанного приложения (руководство пользователя).
- Тестирование.
- Заключение.
- Список использованной литературы.
- Приложение. Листинг разработанной программы.

В разделе «**Постановка задачи**» дается описание решаемой задачи: для чего разрабатывается приложение, какие необходимы входные данные и что является результатом работы программы, какие особенности задачи необходимо учитывать при разработке и использовании программы и т.д.

В разделе «**Описание классов**» для каждого разработанного класса приводится характеристика по плану:

- Назначение класса.
- Данные-элементы класса.
- Операции и функции-утилиты.
- Интерфейс класса (перечисление открытых данных и функций, через которые осуществляется работа с классом). Рекомендуется использовать диаграммы классов UML для описания интерфейса.

В разделе «**Взаимосвязь классов**» необходимо привести схему иерархии наследования и композиции классов.

Если для решения задачи необходимо использовать специальные алгоритмы, например стратегии искусственного интеллекта или методы анализа текстов, то пояснительная записка должна содержать их описание в разделе «**Описание алгоритмов**».

В разделе «**Описание разработанного приложения**» приводятся скриншоты программы и краткая инструкция по использованию.

В пункте «**Тестирование**» приводится контрольный пример, демонстрирующий работоспособность программы. Если в ходе тестирования программы обнаружились несущественные недочеты или ошибки, необходимо их перечислить и привести рекомендации по исправлению.

В **заключении** делаются выводы о проделанной работе (какие приемы ООП были использованы, какие требования из постановки задачи реализованы в программе и т.д.) и пожелания по дальнейшему развитию проекта.

Листинг программы должен иметь необходимые комментарии.

1.2 Оформление отчета

В пояснительной записке к курсовой работе названия разделов и пунктов должны быть выделены жирным шрифтом и размером шрифта. Каждый раздел должен начинаться с новой страницы.

Текст набирается шрифтом Times New Roman 12 или 14 pt. Настройки абзаца: отступ 1,25 см, выравнивание по ширине, междустрочный интервал 1,5 pt, интервал перед и после 0 pt. Заголовки выравниваются по центру.

Страницы должны быть пронумерованы по порядку, номер на титульном листе не проставляется. Поля страницы: слева 3 см, остальные – 1,5 см.

Все таблицы, рисунки и блок-схемы алгоритмов должны быть пронумерованы и подписаны. В тексте необходимо указывать ссылки на них.

1.3 Требования к программе

При написании программы необходимо использовать принципы объектно-ориентированной технологии разработки.

Программа должна быть написана на любом объектно-ориентированном языке и иметь графический интерфейс. Рекомендуется использовать C++ в среде разработке MS Visual Studio, Qt или C++ Builder. Выбор языка программирования, отличного от C++, должен быть согласован с преподавателем.

Для задач из группы «Моделирование» необходимо сохранять параметры модели, чтобы не вводить их при каждом запуске программы.

В программах из группы «Игры» рекомендуется предусмотреть сохранение игры, чтобы продолжить игру при следующем запуске программы.

Задачи из группы «Обработка текстов естественного языка» должны иметь тестовые примеры, которыми проверяется работоспособность программы.

В программах из группы «Информационные и управляющие системы» данные из системы обязательно должны сохраняться для использования при следующем запуске программы.

1.4 Правила сдачи и оценивания курсовой работы

Оценка за курсовую работу проставляется по традиционной шкале и системе ECTS. Оценка определяется исходя из количества набранных баллов по 100-балльной шкале. При сдаче курсовой работы до зачетной недели студенту однократно дается возможность устранить обнаруженные ошибки в программе и отчете. Составляющие оценки соответствуют этапам разработки:

- Анализ проблемы – определение границ решаемой проблемы. Качество проведенного анализа отражается в отчете по курсовой работе, за него проставляется до 20 баллов.
- Проектирование – создание общей структуры системы. Результатом проектирования является набор разработанных классов и их взаимосвязи. Правильно выполненное проектирование оценивается в 40 баллов.
- Реализация – написание и тестирование программы, использующей разработанную систему классов. Программа оценивается в 40 баллов.

При оценке качества отчета по курсовой работе учитываются его содержательная часть и оформление. В случае отсутствия или несоответствия требованиям одного или нескольких пунктов отчета, перечисленных в разделе «1.1 Содержание пояснительной записки», снимается до 20 баллов. Если не учтены требования пункта «1.2 Оформление отчета», снимается до 10 баллов.

Качество проектирования оценивается по оформлению пунктов «Постановка задачи», «Описание классов», «Взаимосвязь классов» и «Описание разработанного приложения» в отчете, по исходному тексту программы и по результатам беседы преподавателя со студентом. До 10 баллов за проектирование назначается за обязательное

использование принципа инкапсуляции и, при необходимости, перегрузки операторов и шаблонов классов. Также необходимо использовать наследование и полиморфизм. За каждый из них можно получить до 10 баллов. В исключительных случаях, если в процессе проектирования выяснилось, что для решения задачи не требуется применять, например, виртуальные функции, и если студент может объяснить свое проектное решение, он также получает до 10 баллов.

Если в процессе работы программы возникают аварийные ситуации (исключения), снимается до 20 баллов в зависимости от ошибки. Наиболее значимыми являются ошибки работы с памятью, возникающие из-за некорректной реализации конструкторов и деструктора. Если не выполнены требования по сохранению данных или отсутствуют тестовые примеры, снимается 10 баллов.

Оставшиеся 20 баллов студент может получить по результатам беседы с преподавателем. Беседа состоит из представления своей работы студентом и ответа на вопросы. Обязательно задаются вопросы по проекту системы и на знание языка программирования.

2. **Критерии оценки.**

Количество баллов за курсовую работу определяется по методике, описанной в разделе 1.4 «Правила сдачи и оценивания курсовой работы»

- работа считается **не выполненной,** если получено от 0 до 20 баллов.
- работа считается выполненной **на пороговом** уровне, если получено от 21 до 50 баллов.
- работа считается выполненной **на базовом** уровне, если получено от 51 до 75 баллов.
- работа считается выполненной **на продвинутом** уровне, если получено от 76 до 100 баллов.

3. **Шкала оценки.**

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

4. **Примерный перечень тем курсового проекта (работы).**

I. Моделирование

1.1. Моделирование работы лифта (здания с пассажирскими и грузовыми лифтами)
1.2. Моделирование перекрестка со светофором
1.3. Моделирование работы автобусного маршрута (включая случайные факторы – поломки автобусов, пробки на дорогах в зависимости от времени)
1.4. Моделирование паромной переправы
1.5. Моделирование производственного конвейера
1.6. Моделирование работы автоматической телефонной станции
1.7. Моделирование работы компании сотовой связи
1.8. Моделирование работы провайдера Интернета
1.9. Моделирование работы аэропорта (управление воздушным движением в зоне аэропорта)
1.10. Моделирование работы аэропорта (обслуживание пассажиров и управление перевозкой грузов)
1.11. Моделирование работы автовокзала
1.12. Моделирование работы метрополитена
1.13. Моделирование работы кафе с официантами
1.14. Моделирование работы прачечной
1.15. Моделирование работы станции автосервиса
1.16. Моделирование работы ремонтной мастерской

1.17. Моделирование работы поликлиники
1.18. Моделирование систем на базе сетей Петри
1.19. Моделирование систем на базе конечных автоматов

II. Игры

2.1. Игра в шахматы
2.2. Игра в домино
2.3. Игра в нарды
2.4. Игра "крестики - нолики" размером 5 х 5 с применением эвристик из искусственного интеллекта
2.5. Имитация настольной игры "Хоккей"
2.6. Имитация настольной игры "Футбол"

Примечание: В программах игр обязательно применять стратегии искусственного интеллекта (альфа-бета отсечение, минимакс и прочие эвристики) для выбора хода компьютера.

III. Обработка текстов естественного языка

3.1. Генератор кроссвордов
3.2. Генератор сканвордов
3.3. Генератор стихотворений
3.4. Генератор басен
3.5. Генератор анекдотов
3.6. Генератор рассказов
3.7. Синтаксический анализатор предложений
3.8. Программа-переводчик с одного языка на другой

IV. Информационные и управляющие системы

4.1. Система складского учета
4.2. Система мониторинга погоды
4.3. АРМ (автоматизированное рабочее место) менеджера по продажам
4.4. АРМ кассира магазина
4.5. АРМ менеджера по персоналу
4.6. АРМ менеджера агентства по подбору кадров
4.7. АРМ менеджера агентства недвижимости
4.8. АРМ регистратора поликлиники
4.9. АРМ врача поликлиники
4.10. АРМ сотрудников деканата
4.11. АРМ сотрудников приемной комиссии ВУЗа
4.12. АРМ оператора компании мобильной связи
4.13. АРМ сотрудников библиотеки
4.14. АРМ сотрудника почтовой службы
4.15. АРМ сотрудника службы курьерской доставки
4.16. АРМ сотрудника бюро расписаний занятий учебного заведения
4.17. АРМ кассира железнодорожного вокзала
4.18. АРМ кассира кинотеатра
4.19. АРМ сотрудника сервис-центра
4.20. Автоматизированная обучающая система
4.21. Система тестирования знаний
4.22. Система дистанционного обучения

5. **Перечень вопросов к защите курсового проекта (работы).**

1. Определение класса и использование классов. Отличия классов и структур.
2. Доступ к членам класса. Функции-утилиты и функции-клиенты класса. Указатель «this».

3. Отделение интерфейса класса от реализации класса.
4. Конструкторы и деструкторы. Вызов конструкторов и деструкторов.
5. Побитовое копирование объекта при присваивании. Конструктор копирования. Правило трех.
6. Передача объектов в функцию по значению, указателю и ссылке. Способы возврата объекта функцией.
7. Динамическое выделение памяти для объекта. Особенности операций new и delete. Массивы объектов. Динамические массивы объектов.
8. Константные объекты и функции-элементы.
9. Композиция.
10. Статические элементы класса.
11. Наследование. Открытые, закрытые и защищенные базовые классы.
12. Переопределение функций базовых классов в производных классах.
13. Конструкторы и деструкторы при наследовании. Вызов конструкторов и деструкторов.
14. Преобразование указателей базовых классов и указателей производных классов.
15. Множественное наследование.
16. Виртуальные функции.
17. Абстрактные базовые классы и конкретные классы.
18. Полиморфизм. Использование полиморфизма.
19. Использование полиморфизма. Статическое и динамическое связывание.
20. Перегрузка бинарных операций. Перегрузка арифметических операций.
21. Перегрузка бинарных операций. Перегрузка оператора присваивания.
22. Перегрузка операций сравнения и логических операций.
23. Перегрузка унарных операций.
24. Дружественные функции в качестве функций-операций.
25. Родовые функции и классы.
26. Перегрузка операций обмена с потоком.
27. Файловые потоки ввода-вывода. Конструирование объекта потока.
28.Обработка исключений. Классы исключений.
29. Исключения и стек. Механизм обработки исключений.