Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Кубанский государственный университет»
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ
Проректор по учебной работе, качеству образования – первый проректор
_____Иванов А.Г.
подпись
«30» 06 2017 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В. ДВ.3.1 ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ МОДЕЛИ

Направление подготовки **01.04.02 Прикладная математика и информатика**

Направленность (профиль) Математическое моделирование

Программа подготовки академическая

Форма обучения очная

Квалификация (степень) выпускника магистр

Краснодар 2017

Рабочая программа дисциплины «ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ МОДЕЛИ» составлена в соответствии с Федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки **01.04.02 Прикладная математика и информатика** (уровень магистратуры), утвержденным приказом Министерства образования и науки Российской Федерации № 911 от 28 августа 2015 г.

Программу составил:

Письменский А.В., канд физ.-мат. наук,
доцент кафедры прикладной математики КубГУ

Рабочая программа дисциплины «Объектно-ориентированные модели» утверждена на заседании кафедры прикладной математики
протокол № 22 « 29 » июня 2017 г.
Заведующий кафедрой прикладной математики
д-р физ.-мат. наук, проф. Уртенов М.Х.

Рабочая программа дисциплины «Объектно-ориентированные модели» утверждена на заседании кафедры математического моделирования
протокол № 16 « 21 » июня 2017 г.
Заведующий кафедрой математического
моделирования акад. РАН,
д-р физ.-мат. наук, проф. Бабешко В.А.

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики
протокол № 4 « 29» июня 2017 г.

Председатель УМК факультета
канд. физ.-мат. наук, доцент Малыхин К.В.

Рецензенты:

Кособуцкая Е.В., канд физ.-мат. наук, доцент, доцент кафедры вычислительных технологий КубГУ.

Марков В.Н. , докт. техн. наук, проф. кафедры информационных систем и программирования института компьютерных систем и информационной безопасности (ИКСиИБ) ФГБОУ ВО «КубГТУ».

1 Цели и задачи учебной дисциплины

1.1 Цель дисциплины

Цели изучения дисциплины определены государственным образовательным стандартом высшего образования и соотнесены с общими целями ООП ВО по направлению подготовки «Прикладная математика и информатика», в рамках которого преподается дисциплина.

Данная дисциплина ставит своей **целью**:

1. Изучение основ классической теории объектно-ориентированного программирования, проектирования и моделирования, в том числе:

- пути эволюции технологий программирования от алгоритмического к объектно-ориентированному (ООП);
- основных принципов объектно-ориентированного построения программных систем (абстракция, инкапсуляция, иерархия, модульность, типизация, параллелизм, сохраняемость);
- понятий классов, объектов, взаимоотношений между ними, а также универсального языка моделирования UML и многоуровневой модели OMG;
- изучение средств объектно-ориентированного и обобщенного программирования языка C++, средств стандартной библиотеки STL.

2. Получение опыта эффективного применения методов объектно-ориентированного программирования и моделирования в научной деятельности, формирование профессиональных навыков исследователя.

Процесс освоения данной дисциплины направлен на получения необходимого объема теоретических знаний, отвечающих требованиям ФГОС ВО и обеспечивающих успешное проведение магистром профессиональной деятельности, владение методологией формулирования и решения прикладных задач, а также на выработку умений применять на практике методы прикладной математики и информатики.

1.2 Задачи дисциплины

1. Усвоение идей и методов объектно-ориентированного моделирования, проектирования и программирования, необходимых для решения теоретических и прикладных задач применения дисциплины.
2. Формирование навыков построения объектно-ориентированных моделей и, в частности, прикладного программного обеспечения (ПО), выбора адекватного математического аппарата их исследования, анализа и практической интерпретации получаемых результатов исследования реальной задачи.
3. Формирование творческого подхода к объектно-ориентированному моделированию и программированию различных процессов; привитие практических навыков использования методов объектно-ориентированного моделирования при решении прикладных задач, анализе и моделировании реальных процессов физики, техники, экологии и др.

1.3 Место дисциплины в структуре образовательной программы

Дисциплина «Объектно-ориентированные модели» относится к вариативной части (Б1.В) учебного плана и является дисциплиной по выбору (ДВ).

Изучение данной учебной дисциплины (модуля) опирается на курсы, аналогичные нижеперечисленным, изучаемым на ступени бакалавриата:

- Алгебра и аналитическая геометрия;
- Основы информатики;
- Языки программирования и методы трансляции;

– Компьютерный практикум;
– Практикум по языкам программирования;
– Базы данных;
– Язык программирования C++ / Программирование на C.

Предварительными требованиями к студентам являются:

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

Перечень последующих учебных дисциплин, для которых необходимы знания, умения и навыки, формируемые данной учебной дисциплиной:

– Современные компьютерные технологии;
– Модели баз данных, насыщенных семантикой;
– Моделирование экологических процессов и систем.

1.4 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся профессиональных компетенций (ПК):

№ п. п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	**уметь**	**владеть**
1.	ПК-3	способностью разрабатывать и применять математические методы, системное и прикладное программное обеспечение для решения задач научной и проектно-технологической деятельности	- основы технологии объектно-ориентированной декомпозиции программных систем, базовых шаблонов проектирования (Наблюдатель, Итератор, Одиночка, Фабрика, Заместитель); - отношения между классами и основы UML (диаграммы классов и последовательностей); - особенности построения объектно-ориентированных программных систем на C++	- строить отношения между классами и строить основные диаграммы UML (диаграммы классов и последовательностей); - создавать объектно-ориентированные программные продукты на языке C++	- основными инструментальными средствами языка C++ и стандартной библиотеки

№ п. п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	**уметь**	**владеть**
2.	ПК-5	способностью управлять проектами, планировать научно-исследовательскую деятельность, анализировать риски, управлять командой проекта	- технологию построения простейших распределенных информационных систем (ИС) и обеспечения безопасности	- управлять проектами; - планировать научно-исследовательскую деятельность	- инструментальными средствами управления проектами

2 Структура и содержание дисциплины

2.1 Распределение трудоемкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 3 зач.ед. (108 часов), их распределение по видам работ представлено в таблице

Вид учебной работы	Трудоемкость, часов
	9 семестр
Аудиторные занятия:	**32**
Занятия лекционного типа (Л)	16
Практические занятия (ПЗ)	–
Лабораторные работы (ЛР)	16
Контролируемая самостоятельная работа (КСР)	–
Самостоятельная работа:	**49**
Курсовой проект (КП), курсовая работа (КР)	–
Проработка учебного (теоретического) материала (ПМ)	29
Подготовка к текущему контролю (ПТК)	20
Реферат (Р)	–
Контролируемая самостоятельная работа (КСР)	–
Иная контактная работа (ИКР)	**0,3**
Промежуточная аттестации (экзамен)	**26,7**
Общая трудоемкость: 3 зач.ед.	**108**

2.2 Структура учебной дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
Разделы дисциплины, изучаемые в 9 семестре

№ раздела	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			Самостоят. работа
			Л	ПЗ	ЛР	СРС
1.	Основные принципы объектно-ориентированного программирования	5	1	–		4

2.	Объектно-ориентированная модель	5	1	–		4
3.	Классы и их описание средствами UML	8	2	–	2	4
4.	Основные методологические отличия C++ от C	6	2	–		4
5.	Средства объектного программирования C++	20	4	–	6	10
6.	Средства объектно-ориентированного программирования C++	13	2	–	2	9
7.	Обобщенное программирование	7	1	–	2	4
8.	Стандартная библиотека C++	9	1	–	2	6
9.	Программирование распределенных приложений	8	2	–	2	4
	Всего по разделам дисциплины:	*81*	*16*	–	*16*	*49*
	ИКР	*0,3*				
	Контроль (экзамен)	*26,7*				
	ИТОГО по дисциплине:	**108**	**16**	–	**16**	**49**

2.3 Содержание разделов дисциплины

№	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
1	Основные принципы объектно-ориентированного программирования	1.1.Эволюция методологий программирования. 1.1.1. Начало начал, или первое поколение языков программирования. 1.1.2. Развитие алгоритмических абстракций, или второе поколение языков программирования. 1.1.3. Модуль как единица построения программных систем, третье поколение языков программирования. 1.1.4. Зарождение объектной модели, четвертое поколение языков программирования. 1.1.5. Объектные языки программирования, объектно-ориентированные языки программирования, объектно-ориентированный анализ, дизайн и проектирование. 1.1.6. Парадигмы программирования. 1.2. Составные части объектного подхода. 1.2.1. Абстрагирование. 1.2.2. Инкапсуляция. 1.2.3. Модульность. 1.2.4. Иерархия. 1.2.5. Типизация. 1.2.6. Параллелизм. 1.2.7. Сохраняемость	Коллоквиум
2	Объектно-ориентированная модель	2.1. Понятие объекта. 2.2. Свойства, присущие объектам. 2.2.1. Состояние. 2.2.2. Поведение. 2.2.3. Идентичность. 2.3.Отношения между объектами. 2.3.1. Типы отношений. 2.3.2. Связь (ассоциация). 2.3.3. Агрегация	Коллоквиум
3	Классы и их описание средствами UML	3.1. Природа классов. 3.2. UML – унифицированный язык моделирования. Четырехуровневая метамодель MOF. 3.3. Отношения между классами. 3.3.1. Типы отношений. 3.3.2. Ассоциация. 3.3.3. Агрегация. 3.3.4. Использование. 3.3.5. Наследование. 3.3.6. Инстанцирование. 3.4. Отношения между классами и объектами	Коллоквиум, защита ЛР

4	Основные методологические отличия C++ от C	4.1. Использование ссылок. Передача аргументов функции по ссылке. 4.2. Использование констант. 4.3. Логические тип и перечисления. 4.4. Операторы управления динамической памятью, инициализация массивов. 4.5. Структура программы, раздельная компиляция и особенности использования статической памяти. 4.6. Пространства имен и исключения (краткий обзор). 4.7. Библиотека ввода вывода (краткий обзор iostream). 4.8. Функциональный полиморфизм	Коллоквиум
5	Средства объектного программирования C++	5.1. Представление объектов и классов. 5.1.1. Реализация поведения объектов на примере добавления функций – членов в структуры. Структура как вырожденный класс. 5.1.2. Структура объявления класса. 5.1.3. Доступ к членам класса. 5.1.4. Поля данных класса как механизм реализации состояния объекта. 5.1.5. Функции члены класса как механизм реализации поведения объекта. 5.1.6. Спецификаторы доступа для обеспечения инкапсуляции. 5.1.7. Средства управления жизнью объекта. Конструкторы и деструкторы. 5.1.8. Конструирование и уничтожение объектов и массивов объектов. 5.1.9. Особенности использования конструктора копии, конструктора по умолчанию, оператора присваивания. 5.1.10. Описание селекторов и модификаторов. 5.1.11. Перегрузка операторов C++ как реализация поведения с предопределенным смыслом. 5.1.12. Дружественность как механизм нарушения инкапсуляции. Достоинства и недостатки механизма дружественности. 5.1.13. Статические поля и методы классов. Инициализация статических полей. 5.2. Реализация отношений между объектами и классами 5.2.1. Ассоциация и агрегация объектов и классов. Зависимость по времени жизни. 5.2.2. Использование и зависимость от интерфейсов. 5.2.3. Объекты при передаче параметров и возврате из методов. 5.2.4. Варианты реализации отношения клиент-сервер. 5.2.5. Внутренние классы	Коллоквиум, защита ЛР, индивидуальные проектные задания

6	Средства объектно-ориентированного программирования C++	6.1. Наследование как средство организации иерархий классов. 6.2. Одиночное наследование. 6.2.1. Понятие производного класса. 6.2.2. Управление доступом в производных классах. 6.2.3. Конструкторы и деструкторы, совмещение имен методов при наследовании, иерархии. 6.2.4. Абстрактные классы и виртуальные функции. 6.2.5. Виртуальный полиморфизм. 6.2.6. Информация о типе на этапе выполнения. RTTI. 6.3. Множественное наследование. 6.3.1. Проблема множественного наследования. Виртуальное наследование как средство разрешения коллизий. 6.3.2. Порядок вызовов конструкторов и деструкторов при множественном наследовании. 6.3.3. Чистые виртуальные классы, понятие интерфейса. 6.3.4. Принципы дизайна иерархий классов. OCP, DIP, ISP. 6.4. Пространства имен. 6.4.1. Пространства имен как средство реализации модульности. 6.4.2. Поиск имен и разрешение конфликтов. 6.4.3. Объединение пространств имен. 6.4.4. Принципы дизайна пакетов. 6.5. Обработка исключений. 6.5.1. Обработка ошибок. 6.5.2. Группировка и перехват исключений. 6.5.3. Управление ресурсами. 6.5.4. Исключения и эффективность. 6.5.5. Альтернативные методы обработки ошибок. 6.5.6. Стандартные исключения.	Коллоквиум, защита ЛР, индивидуальные проектные задания
7	Обобщенное программирование	7.1. Шаблоны классов. 7.1.1. Определение шаблона. 7.1.2. Инстанцирование. 7.1.3. Параметры шаблонов и проверка типов. 7.2. Шаблоны функций. 7.3. Специализация. 7.4. Наследование и шаблоны	Коллоквиум, защита ЛР
8	Стандартная библиотека C++	8.1. Библиотека стандартных шаблонов. 8.1.1. Общие сведения (понятия контейнеров, итераторов и объектов- функций). 8.1.2. Контейнеры (виды контейнеров, последовательные и ассоциативные контейнеры, адаптеры). 8.1.3. Итераторы (итератор как обобщение указателя, классы итераторов). 8.1.4. Алгоритмы (примеры алгоритмов с использованием итераторов: алгоритмы сортировки, алгоритмы, не изменяющие содержание контейнера, алгоритмы, изменяющие содержание контейнера). 8.2. Библиотека ввода-вывода. 8.2.1. Потоки вывода. Вывод типов, определяемых пользователем. 8.2.2. Потоки ввода. Ввод типов, определяемых пользователем. 8.2.3. Форматирование в потоках ввода-вывода. 8.2.4. Буферизация	Коллоквиум, защита ЛР

9	Программирование распределенных приложений	9.1. Принципы построения распределенных приложений. 9.2. Проблемы передачи объектов и синхронизации в распределенных приложениях. 9.3. Реализация сохраняемости. 9.4. Three-tier технология. 9.4.1. Уровень интерфейса. 9.4.2. Уровень бизнес логики. 9.4.3. Уровень сохранения. 9.5. Remote Method Invocation. 9.5.1. Основные принципы и протокол взаимодействия. 9.5.2. Интерфейс Remote и класс UnicastRemoteObject. 9.5.3. Класс Naming и rmiregistry сервис. 9.5.4. RMI сервер. 9.5.5. RMI клиент. 9.5.6. Модель безопасности, синхронизация и сборка мусора в распределенных RMI приложениях. 9.5.7. Механизм Activation. 9.6. Примеры реальных приложений. Сложность разработки	Коллоквиум, защита ЛР, индивидуальные проектные задания

2.3.1 Занятия лекционного типа

№	Наименование раздела	Тема занятия	Форма текущего контроля
1	2	3	4
1.	Основные принципы объектно-ориентированного программирования	1.1.Эволюция методологий программирования. 1.1.1. Начало начал, или первое поколение языков программирования. 1.1.2. Развитие алгоритмических абстракций, или второе поколение языков программирования. 1.1.3. Модуль как единица построения программных систем, третье поколение языков программирования. 1.1.4. Зарождение объектной модели, четвертое поколение языков программирования. 1.1.5. Объектные языки программирования, объектно-ориентированные языки программирования, объектно-ориентированный анализ, дизайн и проектирование. 1.1.6. Парадигмы программирования. 1.2. Составные части объектного подхода. 1.2.1. Абстрагирование. 1.2.2. Инкапсуляция. 1.2.3. Модульность. 1.2.4. Иерархия. 1.2.5. Типизация. 1.2.6. Параллелизм. 1.2.7. Сохраняемость	Коллоквиум
2.	Объектно-ориентированная модель	2.1. Понятие объекта. 2.2. Свойства, присущие объектам. 2.2.1. Состояние. 2.2.2. Поведение. 2.2.3. Идентичность. 2.3.Отношения между объектами. 2.3.1. Типы отношений. 2.3.2. Связь (ассоциация). 2.3.3. Агрегация	Коллоквиум
3.	Классы и их описание средствами UML	3.1. Природа классов. 3.2. UML – унифицированный язык моделирования. Четырехуровневая метамодель MOF. 3.3. Отношения между классами. 3.3.1. Типы отношений. 3.3.2. Ассоциация. 3.3.3. Агрегация. 3.3.4. Использование. 3.3.5. Наследование. 3.3.6. Инстанцирование. 3.4. Отношения между классами и объектами	Коллоквиум
4.	Основные методологические	4.1. Использование ссылок. Передача аргументов функции по ссылке. 4.2. Использование констант. 4.3.	Коллоквиум

№	Наименование раздела	Тема занятия	Форма текущего контроля
1	2	3	4
	отличия C++ от C	Логические тип и перечисления. 4.4. Операторы управления динамической памятью, инициализация массивов. 4.5. Структура программы, раздельная компиляция и особенности использования статической памяти. 4.6. Пространства имен и исключения (краткий обзор). 4.7. Библиотека ввода вывода (краткий обзор iostream). 4.8. Функциональный полиморфизм	
5.	Средства объектного программирования C++	5.1. Представление объектов и классов. 5.1.1. Реализация поведения объектов на примере добавления функций – членов в структуры. Структура как вырожденный класс. 5.1.2. Структура объявления класса. 5.1.3. Доступ к членам класса. 5.1.4. Поля данных класса как механизм реализации состояния объекта. 5.1.5. Функции члены класса как механизм реализации поведения объекта. 5.1.6. Спецификаторы доступа для обеспечения инкапсуляции. 5.1.7. Средства управления жизнью объекта. Конструкторы и деструкторы. 5.1.8. Конструирование и уничтожение объектов и массивов объектов. 5.1.9. Особенности использования конструктора копии, конструктора по умолчанию, оператора присваивания. 5.1.10. Описание селекторов и модификаторов. 5.1.11. Перегрузка операторов C++ как реализация поведения с предопределенным смыслом. 5.1.12. Дружественность как механизм нарушения инкапсуляции. Достоинства и недостатки механизма дружественности. 5.1.13. Статические поля и методы классов. Инициализация статических полей. 5.2. Реализация отношений между объектами и классами 5.2.1. Ассоциация и агрегация объектов и классов. Зависимость по времени жизни. 5.2.2. Использование и зависимость от интерфейсов. 5.2.3. Объекты при передаче параметров и возврате из методов. 5.2.4. Варианты реализации отношения клиент-сервер. 5.2.5. Внутренние классы	Коллоквиум
6.	Средства объектно-ориентированного программирования C++	6.1. Наследование как средство организации иерархий классов. 6.2. Одиночное наследование. 6.2.1. Понятие производного класса. 6.2.2. Управление доступом в производных классах. 6.2.3. Конструкторы и деструкторы, совмещение имен методов при наследовании, иерархии. 6.2.4. Абстрактные классы и виртуальные функции. 6.2.5. Виртуальный полиморфизм. 6.2.6. Информация о типе на этапе выполнения. RTTI. 6.3. Множественное наследование. 6.3.1. Проблема множественного наследования. Виртуальное наследование как средство разрешения коллизий. 6.3.2. Порядок	Коллоквиум

№	Наименование раздела	Тема занятия	Форма текущего контроля
1	2	3	4
		вызовов конструкторов и деструкторов при множественном наследовании. 6.3.3. Чистые виртуальные классы, понятие интерфейса. 6.3.4. Принципы дизайна иерархий классов. OCP, DIP, ISP. 6.4. Пространства имен. 6.4.1. Пространства имен как средство реализации модульности. 6.4.2. Поиск имен и разрешение конфликтов. 6.4.3. Объединение пространств имен. 6.4.4. Принципы дизайна пакетов. 6.5. Обработка исключений. 6.5.1. Обработка ошибок. 6.5.2. Группировка и перехват исключений. 6.5.3. Управление ресурсами. 6.5.4. Исключения и эффективность. 6.5.5. Альтернативные методы обработки ошибок. 6.5.6. Стандартные исключения.	
7.	Обобщенное программирование	7.1. Шаблоны классов. 7.1.1. Определение шаблона. 7.1.2. Инстанцирование. 7.1.3. Параметры шаблонов и проверка типов. 7.2. Шаблоны функций. 7.3. Специализация. 7.4. Наследование и шаблоны	Коллоквиум
8.	Стандартная библиотека C++	8.1. Библиотека стандартных шаблонов. 8.1.1. Общие сведения (понятия контейнеров, итераторов и объектов- функций). 8.1.2. Контейнеры (виды контейнеров, последовательные и ассоциативные контейнеры, адаптеры). 8.1.3. Итераторы (итератор как обобщение указателя, классы итераторов). 8.1.4. Алгоритмы (примеры алгоритмов с использованием итераторов: алгоритмы сортировки, алгоритмы, не изменяющие содержание контейнера, алгоритмы, изменяющие содержание контейнера). 8.2. Библиотека ввода-вывода. 8.2.1. Потоки вывода. Вывод типов, определяемых пользователем. 8.2.2. Потоки ввода. Ввод типов, определяемых пользователем. 8.2.3. Форматирование в потоках ввода-вывода. 8.2.4. Буферизация	Коллоквиум
9.	Программирование распределенных приложений	9.1. Принципы построения распределенных приложений. 9.2. Проблемы передачи объектов и синхронизации в распределенных приложениях. 9.3. Реализация сохраняемости. 9.4. Three-tier технология. 9.4.1. Уровень интерфейса. 9.4.2. Уровень бизнес логики. 9.4.3. Уровень сохранения. 9.5. Remote Method Invocation. 9.5.1. Основные принципы и протокол взаимодействия. 9.5.2. Интерфейс Remote и класс UnicastRemoteObject. 9.5.3. Класс Naming и rmiregistry сервис. 9.5.4. RMI сервер. 9.5.5. RMI клиент. 9.5.6. Модель безопасности, синхронизация и сборка мусора в распределенных RMI приложениях. 9.5.7. Механизм Activation. 9.6. Примеры реальных приложений. Сложность разработки	Коллоквиум

2.3.2 Семинарские занятия

Семинарские занятия не предусмотрены учебным планом.

2.3.3 Лабораторные занятия

№	Наименование раздела	Тема занятия	Форма текущего контроля
1	2	3	4
3.	Классы и их описание средствами UML	1. UML. Диаграммы классов и последовательностей	Защита ЛР
5.	Средства объектного программирования C++	2. Раздельная компиляция и пространства имен. 3. Перегрузка функций, указатели на функции, перечисления. 4. Классы. Перегрузка операций. Реализация матрицы и вектора	Защита ЛР
6.	Средства объектно-ориентированного программирования C++	5. Иерархии классов, наследование. Реализация командного процессора	Защита ЛР
7.	Обобщенное программирование	6. Шаблоны. Реализация шаблонов вектора и матрицы, реализация шаблона «умного» указателя	Защита ЛР
8.	Стандартная библиотека C++	7. Использование библиотеки стандартных шаблонов C++, применение контейнеров	Защита ЛР
9.	Программирование распределенных приложений	8. Распределенное программирование. Сетевое взаимодействие	Защита ЛР

2.3.4 Примерная тематика курсовых работ (проектов)

Курсовые работы не предусмотрены учебным планом.

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающегося по дисциплине

Целью самостоятельной работы студента является углубление знаний, полученных в результате аудиторных занятий. Вырабатываются навыки самостоятельной работы. Закрепляются опыт и знания, полученные во время лабораторных занятий. Ниже представлен перечень учебно-методических материалов, которые помогают обучающемуся организовать самостоятельное изучение тем (вопросов) дисциплины по всем видам СРС.

№	Вид самостоятельной работы	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Проработка и повторение лекционного материала, материала учебной и научной литературы, подготовка к семинарским занятиям	Методические указания для подготовки к лекционным и семинарским занятиям, утвержденные на заседании кафедры прикладной математики факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №22 от 29.06.2017 г. Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры прикладной математики факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №22 от 29.06.2017 г.
2	Подготовка к лабораторным занятиям	Методические указания по выполнению лабораторных работ, утвержденные на заседании кафедры прикладной математики факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №22 от 29.06.2017 г.
3	Подготовка к решению задач и тестов	Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры прикладной математики факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №22 от 29.06.2017 г.
4	Подготовка докладов	Методические указания для подготовки эссе, рефератов, курсовых работ, утвержденные на заседании кафедры прикладной математики факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №22 от 29.06.2017 г.
5	Подготовка к решению расчетно-графических заданий (РГЗ)	Методические указания по выполнению расчетно-графических заданий, утвержденные на заседании кафедры прикладной математики факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №22 от 29.06.2017 г. Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры прикладной математики факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №22 от 29.06.2017 г.
6	Подготовка к текущему контролю	Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры прикладной математики факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №22 от 29.06.2017 г.

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть расширен и конкретизирован в зависимости от контингента обучающихся.

3 Образовательные технологии

Лекционные материалы реализуются с помощью электронных презентаций. При реализации учебной работы по дисциплине «Объектно-ориентированные модели» используются следующие образовательные технологии:

- интерактивная подача материала с мультимедийной системой;
- разбор конкретных исследовательских задач.

Объем интерактивных занятий – 25% от объема аудиторных занятий

Семестр	Вид занятия (Л, ПР, ЛР)	Используемые интерактивные образовательные технологии	Количество часов
4	*Л*	Интерактивная подача материала с мультимедийной системой. Обсуждение сложных и дискуссионных вопросов.	*4*
	ЛР	Компьютерные занятия в режимах взаимодействия «преподаватель - студент».	*4*
ИТОГО			*8*

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

4 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

4.1 Фонд оценочных средств для проведения текущей аттестации

Фонд оценочных средств дисциплины состоит из средств текущего контроля и промежуточной аттестации по итогам освоения дисциплины (экзамена).

В качестве оценочных средств, используемых для текущего контроля успеваемости, предлагается перечень *вопросов по выполненным лабораторным работам*, а также вопросы *коллоквиума*, которые прорабатываются в процессе освоения курса. Данный перечень охватывает все основные разделы курса, включая знания, получаемые для самостоятельной работы. Кроме того, важным элементом обучения является самостоятельное решение студентами и сдача *индивидуальных проектных заданий* в конце курса (на зачетной неделе). Студент демонстрирует свою объектно-ориентированную модель и ее программное решение преподавателю, отвечает на дополнительные вопросы.

4.1.1 Примерные вопросы для коллоквиума

Раздел 1. Основные принципы объектно-ориентированного программирования

1. Эволюция методологий программирования.
2. Первое поколение языков программирования.
3. Развитие алгоритмических абстракций, или второе поколение языков программирования.

4. Модуль как единица построения программных систем, третье поколение языков программирования.
5. Зарождение объектной модели, четвертое поколение языков программирования.
6. Объектные языки программирования, объектно-ориентированные языки программирования, объектно-ориентированный анализ, дизайн и проектирование.
7. Парадигмы программирования.
8. Составные части объектного подхода.
9. Абстрагирование.
10. Инкапсуляция.
11. Модульность.
12. Иерархия.
13. Типизация.
14. Параллелизм.
15. Сохраняемость.

Раздел 2. Объектно-ориентированная модель

1. Понятие объекта.
2. Свойства, присущие объектам.
3. Состояние.
4. Поведение.
5. Идентичность.
6. Отношения между объектами.
7. Типы отношений.
8. Связь (ассоциация).
9. Агрегация.

Раздел 3. Классы и их описание средствами UML

1. Природа классов.
2. UML – унифицированный язык моделирования.
3. Четырехуровневая метамодель MOF.
4. Отношения между классами.
5. Типы отношений.
6. Ассоциация.
7. Агрегация.
8. Использование.
9. Наследование.
10. Инстанцирование.
11. Отношения между классами и объектами.

Раздел 4. Основные методологические отличия C++ от C

1. Использование ссылок. Передача аргументов функции по ссылке.
2. Использование констант.
3. Логические тип и перечисления.
4. Операторы управления динамической памятью, инициализация массивов.
5. Структура программы, раздельная компиляция и особенности использования статической памяти.
6. Пространства имен и исключения (краткий обзор).
7. Библиотека ввода вывода (краткий обзор iostream).
8. Функциональный полиморфизм.

Раздел 5. Средства объектного программирования C++

1. Представление объектов и классов.
2. Реализация поведения объектов на примере добавления функций – членов в структуры.
3. Структура как вырожденный класс.

4. Структура объявления класса.
5. Доступ к членам класса.
6. Поля данных класса как механизм реализации состояния объекта.
7. Функции члены класса как механизм реализации поведения объекта.
8. Спецификаторы доступа для обеспечения инкапсуляции.
9. Средства управления жизнью объекта.
10. Конструкторы и деструкторы.
11. Конструирование и уничтожение объектов и массивов объектов.
12. Особенности использования конструктора копии, конструктора по умолчанию, оператора присваивания.
13. Описание селекторов и модификаторов.
14. Перегрузка операторов C++ как реализация поведения с предопределенным смыслом.
15. Дружественность как механизм нарушения инкапсуляции.
16. Достоинства и недостатки механизма дружественности.
17. Статические поля и методы классов. Инициализация статических полей.
18. Реализация отношений между объектами и классами.
19. Ассоциация и агрегация объектов и классов. Зависимость по времени жизни.
20. Использование и зависимость от интерфейсов.
21. Объекты при передаче параметров и возврате из методов.
22. Варианты реализации отношения клиент-сервер.
23. Внутренние классы.

Раздел 6. Средства объектно-ориентированного программирования C++

1. Наследование как средство организации иерархий классов.
2. Одиночное наследование.
3. Понятие производного класса.
4. Управление доступом в производных классах.
5. Конструкторы и деструкторы, совмещение имен методов при наследовании, иерархии.
6. Абстрактные классы и виртуальные функции.
7. Виртуальный полиморфизм.
8. Информация о типе на этапе выполнения. RTTI. Множественное наследование.
9. Проблема множественного наследования.
10. Виртуальное наследование как средство разрешения коллизий.
11. Порядок вызовов конструкторов и деструкторов при множественном наследовании.
12. Чистые виртуальные классы, понятие интерфейса.
13. Принципы дизайна иерархий классов. OCP, DIP, ISP.
14. Пространства имен.
15. Пространства имен как средство реализации модульности.
16. Поиск имен и разрешение конфликтов.
17. Объединение пространств имен.
18. Принципы дизайна пакетов.
19. Обработка исключений.
20. Обработка ошибок.
21. Группировка и перехват исключений.
22. Управление ресурсами.
23. Исключения и эффективность.
24. Альтернативные методы обработки ошибок.
25. Стандартные исключения.

Раздел 7. Обобщенное программирование

1. Шаблоны классов.
2. Определение шаблона.
3. Инстанцирование.
4. Параметры шаблонов и проверка типов.
5. Шаблоны функций.
6. Специализация.
7. Наследование и шаблоны.

Раздел 8. Стандартная библиотека C++

1. Библиотека стандартных шаблонов.
2. Общие сведения (понятия контейнеров, итераторов и объектов- функций).
3. Контейнеры (виды контейнеров, последовательные и ассоциативные контейнеры, адаптеры).
4. Итераторы (итератор как обобщение указателя, классы итераторов).
5. Алгоритмы (примеры алгоритмов с использованием итераторов: алгоритмы сортировки, алгоритмы, не изменяющие содержание контейнера, алгоритмы, изменяющие содержание контейнера).
6. Библиотека ввода-вывода.
7. Потоки вывода. Вывод типов, определяемых пользователем.
8. Потоки ввода. Ввод типов, определяемых пользователем.
9. Форматирование в потоках ввода-вывода.
10. Буферизация.

Раздел 9. Программирование распределенных приложений

1. Принципы построения распределенных приложений.
2. Проблемы передачи объектов и синхронизации в распределенных приложениях.
3. Реализация сохраняемости.
4. Three-tier технология.
5. Уровень интерфейса.
6. Уровень бизнес логики.
7. Уровень сохранения.
8. Remote Method Invocation.
9. Основные принципы и протокол взаимодействия.
10. Интерфейс Remote и класс UnicastRemoteObject.
11. Класс Naming и rmiregistry сервис.
12. RMI сервер.
13. RMI клиент.
14. Модель безопасности, синхронизация и сборка мусора в распределенных RMI приложениях.
15. Механизм Activation.
16. Сложность разработки.

4.1.2 Примерные задания для индивидуальных проектов

1. Используя основные графические объекты MS VS C++, разработать *однодокументный* (SDI) растровый графический редактор с возможностью сохранения изображений в форматах BMP и GIF.
2. Используя основные графические объекты MS VS C++, разработать *многодокументный* (MDI) растровый графический редактор с возможностью сохранения изображений в любом из стандартных графических форматов.
3. Разработать векторный графический редактор.
4. Разработать WFA-программу для преобразования чисел между основными системами счисления: 2-чной (Bin), 8-чной (Oct), 10-чной (Dec), 16-чной (Hex), аналогично соответствующей функции стандартного калькулятора Windows.

5. Разработать WFA-программу проверки знаний по данной дисциплине специализации.
6. Разработать WFA-программу, реализующую IQ-тестирование.
7. Разработать WFA-программу, выполняющую психологическое тестирование.
8. Разработать WFA-программу, выполняющую обработку результатов общественного мнения.
9. Разработать WFA-программу приготовления коктейлей. Классы напитков описать в отдельном модуле.
10. Разработать игру "Однорукий бандит". Использовать компонент TImageList.
11. Разработать игру "Крестики-нолики".
12. Разработать игру "Сапёр".
13. Разработать игру "Тетрис".
14. Разработать игру "Морской бой".
15. Разработать WFA-программу, выполняющую *пузырьковую* сортировку данных. Предусмотреть диалоги загрузки исходных данных из файла и записи результата в файл.
16. Разработать WFA-программу, выполняющую сортировку данных *перемешиванием* (шейкером). Предусмотреть диалоги загрузки исходных данных из файла и записи результата в файл.
17. Разработать WFA-программу, выполняющую сортировку данных *методом вставок*. Предусмотреть диалоги загрузки исходных данных из файла и записи результата в файл.
18. Разработать WFA-программу, выполняющую сортировку данных *подсчётом*. Предусмотреть диалоги загрузки исходных данных из файла и записи результата в файл.
19. Разработать WFA- программу, выполняющую сортировку данных *методом Шелла*. Предусмотреть диалоги загрузки исходных данных из файла и записи результата в файл.
20. Разработать WFA-программу, реализующую метод *быстрой* сортировки данных. Предусмотреть диалоги загрузки исходных данных из файла и записи результата в файл.
21. Разработать WFA-программу, выполняющую операции над матрицами: *умножение на вектор*, *перемножение матриц*. Матрицы задавать на форме с помощью компонента TStringGrid. Предусмотреть диалоги загрузки исходных данных из файла и записи результата в файл.
22. Разработать WFA-программу, выполняющую *обращение* матрицы. Матрицу задавать на форме с помощью компонента TStringGrid. Предусмотреть диалоги загрузки исходных данных из файла и записи результата в файл.
23. Разработать WFA-программу, выполняющую *вычисление определителя* матрицы. Матрицу задавать на форме с помощью компонента TStringGrid. Предусмотреть диалоги загрузки исходных данных из файла и записи результата в файл.
24. Разработать WFA-программу, выполняющую *проверку знакоопределенности матрицы по критерию Сильвестра*. Матрицу задавать на форме с помощью компонента TStringGrid. Предусмотреть диалоги загрузки исходных данных из файла и записи результата в файл.

4.2 Фонд оценочных средств для проведения промежуточной аттестации

4.2.1 Примерный перечень вопросов к экзамену

Общая теория ООП:

1. Эволюция методологий программирования. Парадигмы программирования.
2. Основные принципы объектного подхода. Абстрагирование.

3. Основные принципы объектного подхода. Инкапсуляция.
4. Основные принципы объектного подхода. Модульность.
5. Основные принципы объектного подхода. Иерархия.
6. Основные принципы объектного подхода. Типизация.
7. Основные принципы объектного подхода. Параллелизм. Сохраняемость.
8. Объект с точки зрения ООП. Состояние. Поведение.
9. Объект с точки зрения ООП. Идентичность и жизненный цикл объектов.
10. Объект с точки зрения ООП. Взаимоотношения между объектами.
11. Классы. Природа классов. Метамодель. Инстанцирование.
12. Классы. Структура класса. Абстрактные классы и интерфейсы.
13. Классы. Отношения между классами. Ассоциация и агрегация.
14. Классы. Иерархии классов. Зависимость.

Средства C++:

15. Модель памяти и структура программы. Классы памяти. Ссылки.
16. Средства абстракции C++. Структура класса. Статические члены.
17. Средства инкапсуляции C++. Инкапсуляция и наследование. Друзья.
18. Модульность, раздельная компиляция, пространства имен, директива using.
19. Представление иерархических отношений. Наследование.
20. Представление иерархических отношений. Агрегация. Зависимость по времени жизни.
21. Правила преобразования типов в C++. Параметрический и виртуальный полиморфизм.
22. C++: средства реализации состояния объектов; реализация поведения.
23. Перегрузка операторов.
24. Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании.
25. Варианты реализации отношения клиент-сервер. Объекты при передаче параметров и возврате из методов.
26. Исключения в C++. Обработка исключений.
27. Шаблоны классов и шаблоны функций. Специализация.
28. Основы STL. Структура и назначение. Контейнеры. Алгоритмы.
29. Стандартная библиотека, ввод-вывод.

Программирование распределенных приложений

30. Принципы построения распределенных приложений.
31. Проблемы передачи объектов и синхронизации в распределенных приложениях.
32. Реализация сохраняемости.
33. Three-tier технология.
34. Remote Method Invocation (RMI).

4.2.2 Критерии оценки

Оценка «отлично»:

– систематизированные, глубокие и полные знания по всем разделам дисциплины, а также по основным вопросам, выходящим за пределы учебной программы;

– точное использование научной терминологии систематически грамотное и логически правильное изложение ответа на вопросы;

– безупречное владение инструментарием учебной дисциплины, умение его эффективно использовать в постановке научных и практических задач;

– выраженная способность самостоятельно и творчески решать сложные проблемы и нестандартные ситуации;

– полное и глубокое усвоение основной и дополнительной литературы, рекомендованной учебной программой по дисциплине;

– умение ориентироваться в теориях, концепциях и направлениях дисциплины и давать им критическую оценку, используя научные достижения других дисциплин;

– творческая самостоятельная работа на практических/семинарских/лабораторных занятиях, активное участие в групповых обсуждениях, высокий уровень культуры исполнения заданий;

– высокий уровень сформированности заявленных в рабочей программе компетенций.

Оценка «хорошо»:

– достаточно полные и систематизированные знания по дисциплине;

– умение ориентироваться в основном теориях, концепциях и направлениях дисциплины и давать им критическую оценку;

– использование научной терминологии, лингвистически и логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;

– владение инструментарием по дисциплине, умение его использовать в постановке и решении научных и профессиональных задач;

– усвоение основной и дополнительной литературы, рекомендованной учебной программой по дисциплине;

– самостоятельная работа на практических занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий;

– средний уровень сформированности заявленных в рабочей программе компетенций.

Оценка «удовлетворительно»:

– достаточный минимальный объем знаний по дисциплине;

– усвоение основной литературы, рекомендованной учебной программой;

– умение ориентироваться в основных теориях, концепциях и направлениях по дисциплине и давать им оценку;

– использование научной терминологии, стилистическое и логическое изложение ответа на вопросы, умение делать выводы без существенных ошибок;

– владение инструментарием учебной дисциплины, умение его использовать в решении типовых задач;

– умение под руководством преподавателя решать стандартные задачи;

– работа под руководством преподавателя на практических занятиях, допустимый уровень культуры исполнения заданий;

– достаточный минимальный уровень сформированности заявленных в рабочей программе компетенций.

Оценка «неудовлетворительно»:

– фрагментарные знания по дисциплине;

– отказ от ответа (выполнения письменной работы);

– знание отдельных источников, рекомендованных учебной программой по дисциплине;

– неумение использовать научную терминологию;

– наличие грубых ошибок;

– низкий уровень культуры исполнения заданий;

– низкий уровень сформированности заявленных в рабочей программе компетенций.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть дополнен и конкретизирован в зависимости от контингента обучающихся.

5 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

5.1 Основная литература

1. Боггс У., Боггс М. UML и Rational Rose: секреты эффективного проектирования сопровождаемых объектно-ориентированных приложений / пер. И. Афанасьев, И. Дранишников. — М: ЛОРИ, 2008. — 580 с.
2. Павловская Т.А. C/C++. Программирование на языке высокого уровня: для магистров и бакалавров: учебник для студентов вузов. — СПб: Питер, 2014. — 460 с.
3. Ашарина И.В. Объектно-ориентированное программирование в C++: лекции и упражнения. [Электронный ресурс]. — М.: Горячая линия-Телеком, 2012. — 320 с. — Режим доступа: http://e.lanbook.com/book/5115.
4. Лаврищева Е.М. Программная инженерия. Парадигмы, технологии и case-средства: учебник для вузов [Электронный ресурс]. — 2-е изд., испр. — М.: Издательство Юрайт, 2017. — 280 с. — (Университеты России). — Режим доступа: https://www.biblio-online.ru/book/DCE62C40-BE54-4478-9BA5-7BE6200A8967.

Для освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья имеются издания в электронном виде в электронно-библиотечных системах «Лань» и «Юрайт».

5.2 Дополнительная литература

1. Иванов Д. Моделирование на UML. [Электронный ресурс] / Д. Иванов, Ф. Новиков. — Электрон. дан. — СПб. : НИУ ИТМО, 2010. — 200 с. — Режим доступа: http://e.lanbook.com/book/40879 — Загл. с экрана.
2. Кватрани, Т. Rational Rose 2000 и UML. Визуальное моделирование. [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2009. — 176 с. — Режим доступа: http://e.lanbook.com/book/1237 — Загл. с экрана.

3. Розенберг, Д. Применение объектного моделирования с использованием UML и анализ прецедентов. [Электронный ресурс] / Д. Розенберг, К. Скотт. — Электрон. дан. — М. : ДМК Пресс, 2007. — 160 с. — Режим доступа: http://e.lanbook.com/book/1226 — Загл. с экрана.
4. Иванова Г.С. Объектно-ориентированное программирование / Г.С. Иванова, Т.Н. Ничушкина, Е.К. Пугачев ; под ред. Г. С. Ивановой. — М.: Изд-во МГТУ им. Н. Э. Баумана, 2001. — 317 с.
5. Лаптев В.В. С++. Объектно-ориентированное программирование: задачи и упражнения: учебное пособие для студентов вузов / В. В. Лаптев, А. В. Морозов, А. В. Бокова. — СПб. [и др.] : ПИТЕР , 2007. — 287 с.
6. Страуструп Б. Язык программирования С++ / Пер. с англ. — М.: Изд-во БИНОМ, 2004. — 1098 с.
7. Павловская Т.А. C#. Программирование на языке высокого уровня. — СПб: Питер, 2014. — 432 с.
8. Литвиненко Н.А. Технология программирования на С++. Win32 API-приложения [Электронный ресурс]. — СПб.: БХВ-Петербург, 2010. — 280 с. — Режим доступа: http://znanium.com.
9. Культин Н.Б. Основы программирования в Microsoft Visual C++ 2010 [Электронный ресурс]. — СПб.: БХВ–Петербург, 2010. — 384 с. — Режим доступа: http://ibooks.ru.
10. Культин Н.Б. Основы программирования в Microsoft Visual C# 2010 [Электронный ресурс]. — СПб.: БХВ–Петербург, 2011. — 368 с. — Режим доступа: http://ibooks.ru.

6 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. IBM Rational Rose Modeler [Электронный ресурс]. — URL: http://www-03.ibm.com/software/products/ru/rosemod.
2. Object Management Group – Теория объектного подхода, UML [Электронный ресурс]. — URL: http://www.omg.org.
3. Информация по языку и стандартной библиотеке С++ [Электронный ресурс]. — URL: http://cplusplus.com.
4. Microsoft Developer Network [Электронный ресурс]. — URL: http://msdm.microsoft.com.
5. Википедия, свободная энциклопедия – Wikipedia [Электронный ресурс]. — URL: http://ru.wikipedia.org.
6. Электронно-библиотечная система Издательство «Лань» [Электронный ресурс]. — URL: http://e.lanbook.com.
7. Научная электронная библиотека в рамках проекта Федерального агентства по науке и инновациям [Электронный ресурс]. — URL: elibrary.ru.

7 Методические рекомендации по организации изучения дисциплины

Учебная деятельность проходит в соответствии с графиком учебного процесса. Процесс самостоятельной работы контролируется во время аудиторных занятий и индивидуальных консультаций. Самостоятельная работа студентов проводится в форме изучения отдельных теоретических вопросов по предлагаемой литературе.

По желанию студента предлагается написание реферата на выбранную им тему (по согласованию с преподавателем). Для написания реферата необходимо подобрать литературу. Общее количество литературных источников, включая тексты из Интернета, (публикации в журналах), должно составлять не менее 10 наименований. Учебники, как правило, в литературные источники не входят.

Рефераты выполняют на листах формата А4. Страницы текста, рисунки, формулы нумеруют, рисунки снабжают подрисуночными надписями. Текст следует печатать шрифтом №14 с интервалом между строками в 1,5 интервала, без недопустимых сокращений. В конце реферата должны быть сделаны выводы.

В конце работы приводят список использованных источников.

Реферат должен быть подписан студентом с указанием даты ее оформления.

Работы, выполненные без соблюдения перечисленных требований, возвращаются на доработку.

Выполненная студентом работа определяется на проверку преподавателю в установленные сроки. Если у преподавателя есть замечания, работа возвращается и после исправлений либо вновь отправляется на проверку, если исправления существенные, либо предъявляется на ее защите.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

8 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю) (при необходимости)

8.1 Перечень информационных технологий

– Проверка домашних заданий и консультирование посредством электронной почты и социальной сети «ВКонтакте».

– Использование электронных презентаций при проведении лекционных и лабораторных занятий.

8.2 Перечень необходимого программного обеспечения

1. Операционная система MS Windows.
2. Интегрированное офисное приложение MS Office.
3. Программное обеспечение для организации управляемого коллективного и безопасного доступа в Интернет.
4. Система программирования MS Visual Studio или Borland C++ Builder.

8.3 Перечень необходимых информационных справочных систем

1. Информация по языку и стандартной библиотеке C++ [Электронный ресурс]. — URL: http://cplusplus.com.
2. Википедия, свободная энциклопедия – Wikipedia [Электронный ресурс]. — URL: http://ru.wikipedia.org.
3. Электронная библиотека КубГУ [Электронный ресурс]. — URL: http://www.kubsu.ru/ru/node/1145.

9 Материально-техническое обеспечение учебной дисциплины

№	Вид работ	Материально-техническое обеспечение дисциплины и оснащенность
1.	Лекционные занятия	Лекционная аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук), соответствующим программным обеспечением, а также необходимой мебелью (доска, столы, стулья) (аудитории: 129, 131, 133, А305, А307)
2.	Лабораторные занятия	Лаборатория, укомплектованная специализированной мебелью, техническими средствами обучения (современными ПЭВМ на базе процессоров Intel или AMD, объединёнными локальной сетью) с выходом в глобальную сеть Интернет, а также современным лицензионным программным обеспечением (операционная система Windows 8/10, пакет Microsoft Office, среды программирования MS Visual Studio и Delphi) (аудитории: 101, 102, 105, 106, 107, А301а)
3.	Групповые (индивидуальные) консультации	Аудитория для семинарских занятий, групповых и индивидуальных консультаций, укомплектованные необходимой мебелью (доска, столы, стулья) (аудитории: 129, 131)
4.	Текущий контроль, промежуточная аттестация	Аудитория для семинарских занятий, текущего контроля и промежуточной аттестации, укомплектованная необходимой мебелью (доска, столы, стулья) (аудитории: 129, 131, 133, А305, А307, 147, 148, 149, 150, 100С, А3016, А512), компьютерами с лицензионным программным обеспечением и выходом в интернет (аудитории: 106, 106а. А301)
5.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети Интернет, программой экранного увеличения, обеспеченный доступом в электронную информационно-образовательную среду университета, необходимой мебелью (доска, столы, стулья) (аудитория 102а, читальный зал).