

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего обра-
зования «Кабардино-Балкарский государственный университет
им. Х.М. Бербекова» (КБГУ)

ИНСТИТУТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И ЦИФРОВЫХ ТЕХНОЛОГИЙ
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

СОГЛАСОВАНО

Руководитель образовательной
программы *М.Р. Яхутлова* М.Р. Яхутлова
« *02* » *09* 2022г.

УТВЕРЖДАЮ
Директор института
А.Х. Шапсигов А.Х. Шапсигов
« *02* » *09* 2022г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

01.03.02. Прикладная математика и информатика
(код и наименование направления подготовки)

Математическое и компьютерное моделирование
(наименование профиля подготовки)

Квалификация (степень) выпускника
Бакалавр

Форма обучения
Очная

Нальчик 2022

Рабочая программа дисциплины «Объектно-ориентированное программирование» / сост. А.Л. Нагоров – Нальчик: КБГУ, 2022. – 36 с.

Рабочая программа предназначена для студентов очной формы обучения, по направлению подготовки 01.03.02 Прикладная математика и информатика профиль «Математическое и компьютерное моделирование» V семестра, 3 курса.

Рабочая программа составлена с учетом Федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки 01.03.02 Прикладная математика и информатика (уровень бакалавриата), утвержденного и введенного в действие приказом Минобрнауки России № 9 от 10.01.2018.

Содержание

1. Цель и задачи освоения дисциплины	4
2. Место дисциплины в структуре ОПОП ВО	4
3. Требования к результатам освоения дисциплины.....	4
4. Содержание и структура дисциплины (модуля).....	5
5. Оценочные материалы для текущего и рубежного контроля успеваемости	9
и промежуточной аттестации	9
6. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности	24
7. Учебно-методическое обеспечение дисциплины	26
7.1 Основная литература	26
7.2 Дополнительная литература	27
7.3 Интернет-ресурсы.....	27
7.4 Методические указания по проведению различных учебных занятий,	30
к курсовому проектированию и другим видам самостоятельной работы.	30
8. Материально-техническое обеспечение дисциплины	34
9. Лист изменений (дополнений)	36
10. Приложения.....	37

1. Цель и задачи освоения дисциплины

Целями освоения дисциплины «Объектно-ориентированное программирование» являются изучение основ классической теории объектно-ориентированного программирования и средств объектно-ориентированного и обобщенного программирования на языке C++.

Задачами освоения учебной дисциплины являются

- формирование представлений об общей методологии и средствах технологии объектно-ориентированного программирования;
- углубленная подготовка студентов в области применения технологии объектно-ориентированного программирования.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Объектно-ориентированное программирование» относится к обязательной части Блока 1 по направлению подготовки «Прикладная математика и информатика». Дисциплина изучается в 5 семестре и предъявляет требования к «входным» знаниям, умениям обучающегося в области программирования и опирается на дисциплины «Языки и методы программирования», «Практикум на ЭВМ», «Основы информатики».

3. Требования к результатам освоения дисциплины

В результате освоения дисциплины формируются следующие компетенции выпускника согласно ФГОС ВО:

ПКС-2 Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности.

В результате освоения дисциплины обучающийся должен:

Знать:

- концепцию ООП
- правила составления программ на языке программирования C++
- основные возможности сред программирования DEV-C++, Microsoft Visual C++.

Уметь:

- составлять программы на языке программирования C++
- использовать среду программирования DEV-C++, Microsoft Visual C++ для разработки и отладки программ на языке C++

Владеть:

- методами ООП
- инструментальными средствами ООП
- методами отладки программных продуктов.

4. Содержание и структура дисциплины (модуля)

Таблица 1. Содержание дисциплины (модуля) «Объектно-ориентированное программирование», перечень оценочных средств и контролируемых компетенций (ПКС-2)

№	Наименование раздела/темы	Содержание раздела	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Основные принципы объектно-ориентированного программирования	1.1. Эволюция методологий программирования Начало начал, или первое поколение языков программирования. Развитие алгоритмических абстракций, или второе поколение языков программирования. Модуль как единица построения программных систем, третье поколение языков программирования. Зарождение объектной модели, четвертое поколение языков программирования. Объектные языки программирования, объектно-ориентированные языки программирования, объектно-ориентированный анализ, дизайн и проектирование. Парадигмы программирования. 1.2. Составные части объектного подхода Абстрагирование. Инкапсуляция. Модульность. Иерархия. Типизация. Параллелизм. Сохраняемость	ПКС-2	Коллоквиум (К), рубежный контроль (РК), тестирование (Т)
2	Объектно-ориентированная модель	2.1. Понятие объекта 2.2. Свойства, присущие объектам Состояние. Поведение. Идентичность. Отношения между объектами. Типы отношений. Связь (ассоциация). Агрегация	ПКС-2	Коллоквиум (К), рубежный контроль (РК), тестирование (Т)
3	Классы	3.1. Природа классов. 3.2. UML – унифицированный язык моделирования. Четырехуровневая метамодель MOF 3.3. Отношения между классами. Типы отношений. Ассоциация. Агрегация. Использование. Наследование. Инстанцирование 3.4. Отношения между классами и объектами	ПКС-2	Коллоквиум (К), рубежный контроль (РК), тестирование (Т)
4	Основные алгоритмические отличия C++ от C	4.1. Использование ссылок. Передача аргументов функции по ссылке. 4.2. Использование констант. 4.3. Логические тип и перечисления. 4.4. Операторы управления динамической памятью, инициализация массивов. 4.5. Структура программы, отдельная 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.	ПКС-2	Коллоквиум (К), рубежный контроль (РК), тестирование (Т)

5	Средства объектного программирования языка C++	<p>5.1. Представление объектов и классов. Реализация поведения объектов на примере добавления функций—членов в структуры. Структура как вырожденный класс. Структура объявления класса. Доступ к членам класса. Поля данных класса как механизм реализации состояния объекта. Функции члены класса как механизм реализации поведения объекта. Спецификаторы доступа для обеспечения инкапсуляции. Средства управления жизнью объекта. Конструкторы и деструкторы. Конструирование и уничтожение объектов и массивов объектов. Особенности использования конструктора копии, конструктора по умолчанию, оператора присваивания. Описание селекторов и модификаторов. Перегрузка операторов C++ как реализация поведения с предопределенным смыслом. Дружественность как механизм нарушения инкапсуляции. Достоинства и недостатки механизма дружественности. Статические поля и методы классов. Инициализация статических полей.</p> <p>5.2. Реализация отношений между объектами и классами Ассоциация и агрегация объектов и классов. Зависимость по времени жизни. Использование и зависимость от интерфейсов. Объекты при передаче параметров и возврате из методов. Варианты реализации отношения клиент-сервер. Внутренние классы</p>	ПКС-2	Коллоквиум (К), рубежный контроль (РК), тестирование (Т)
6	Средства объектно-ориентированного программирования C++	<p>6.1. Наследование как средство организации иерархий классов. Принцип замещения Лисковской.</p> <p>6.2. Одиночное наследование. Понятие производного класса. Управление доступом в производных классах. Конструкторы и деструкторы, совмещение имен методов при наследовании, иерархии. Абстрактные классы и виртуальные функции. Виртуальный полиморфизм. Информация о типе на этапе выполнения. RTTI.</p> <p>6.3. Множественное наследование Проблема множественного наследования. Виртуальное наследование как средство разрешения коллизий. Порядок вызовов конструкторов и деструкторов при множественном наследовании. Чистые виртуальные классы, понятие интерфейса. Принципы дизайна иерархий классов. OCP, DIP, ISP.</p> <p>6.4. Пространства имен. 6.4.1. Пространства имен как средство реализации модульности. Поиск имен и разрешение конфликтов. Объединение пространств имен. Принципы дизайна пакетов.</p> <p>6.5. Обработка исключений. Обработка ошибок. Группировка и перехват исключений. Управление ресурсами. Исключения и эффективность. Альтернативные методы обработки ошибок. Стандартные исключения.</p>	ПКС-2	Коллоквиум (К), рубежный контроль (РК), тестирование (Т)
7	Обобщенное программирование.	<p>7.1. Шаблоны классов. Определение шаблона. Инстанцирование. Параметры шаблонов и проверка типов.</p> <p>7.2. Шаблоны функций.</p> <p>7.3. Специализация.</p> <p>7.4. Наследование и шаблоны.</p>	ПКС-2	Коллоквиум (К), рубежный контроль (РК), тестирование (Т)
8	Стандартная библиотека C++	<p>8.1. Библиотека стандартных шаблонов Общие сведения (понятия контейнеров, итераторов и объектов-функций) Контейнеры (виды контейнеров, последовательные и ассоциативные контейнеры, адаптеры) Итераторы (итератор как обобщение указателя, классы итераторов). Алгоритмы (примеры алгоритмов с использованием итераторов: алгоритмы сортировки, алгоритмы, не изменяющие содержание контейнера, алгоритмы, изменяющие содержание контейнера)</p> <p>8.2. Библиотека ввода-вывода Потоки вывода. Вывод типов определяемых пользователем. Потоки ввода. Ввод типов определяемых пользователем. Форматирование в потоках ввода-вывода. Буферизация.</p>	ПКС-2	Коллоквиум (К), рубежный контроль (РК), тестирование (Т)

Структура дисциплины (модуля) «Объектно-ориентированное программирование»

Таблица 2. Общая трудоемкость дисциплины составляет 3 зачетные единицы (108 часов)

Вид работы	Трудоемкость, часов / зачетных единиц	
	Всеместр	всего
Общая трудоемкость (в часах)	108	108
Контактная работа (в часах):	34	34
Лекционные занятия (Л)	17	17
Практические занятия (ПЗ)		
Семинарские занятия (СЗ)		
Лабораторные работы (ЛР)	17	17
Самостоятельная работа (в часах), в том числе контактная работа:	47	47
Расчетно-графическое задание		
Реферат (Р)		
Эссе (Э)		
Контрольная работа (КР)		
Самостоятельное изучение разделов		
Курсовой проект (КП), курсовая работа (КР)		
Подготовка и прохождение промежуточной аттестации	27	27
Вид промежуточной аттестации	экзамен	экзамен

Таблица 3. Лекционные занятия

№	Наименование темы
1	Основные принципы объектно-ориентированного программирования
2	Объектно-ориентированная модель
3	Классы
4	Основные алгоритмические отличия C++ от C
5	Средства объектного программирования языка C++
6	Средства объектно-ориентированного программирования C++
7	Обобщенное программирование.
8	Стандартная библиотека C++

Таблица 4. Практические занятия (семинарские занятия) – не предусмотрены

Таблица 5. Лабораторные работы по дисциплине (модулю)

№ занятия	Наименование темы
1	Классы и объекты в C++
2	Наследование и виртуальные функции
3	Иерархия объектов и группа. Итераторы.
4	Обработка событий
5	Перегрузка операций
6	Шаблоны функций и классов
7	Потоковые классы
8	Стандартная библиотека шаблонов

Таблица 6. Самостоятельное изучение разделов дисциплины (модуля)

№ разде-	Вопросы, выносимые на самостоятельное изучение
----------	--

ла	
1	<p>1.1. Эволюция методологий программирования Начало начал, или первое поколение языков программирования. Развитие алгоритмических абстракций, или второе поколение языков программирования. Модуль как единица построения программных систем, третье поколение языков программирования. Зарождение объектной модели, четвертое поколение языков программирования. Объектные языки программирования, объектно-ориентированные языки программирования, объектно-ориентированный анализ, дизайн и проектирование. Парадигмы программирования.</p> <p>1.2. Составные части объектного подхода Абстрагирование. Инкапсуляция. Модульность. Иерархия. Типизация. Параллелизм. Сохраняемость</p>
2	<p>2.1. Понятие объекта 2.2. Свойства, присущие объектам Состояние. Поведение. Идентичность. Отношения между объектами. Типы отношений. Связь (ассоциация). Агрегация</p>
3	<p>3.1. Природа классов. 3.2. UML – унифицированный язык моделирования. Четырехуровневая метамодель MOF 3.3. Отношения между классами. Типы отношений. Ассоциация. Агрегация. Использование. Наследование. Инстанцирование 3.4. Отношения между классами и объектами</p>
4	<p>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, отдельная 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.</p>
5	<p>5.1. Представление объектов и классов. Реализация поведения объектов на примере добавления функций—членов в структуры. Структура как вырожденный класс. Структура объявления класса. Доступ к членам класса. Поля данных класса как механизм реализации состояния объекта. Функции члены класса как механизм реализации поведения объекта. Спецификаторы доступа для обеспечения инкапсуляции. Средства управления жизнью объекта. Конструкторы и деструкторы. Конструирование и уничтожение объектов и массивов объектов. Особенности использования конструктора копии, конструктора по умолчанию, оператора присваивания. Описание селекторов и модификаторов. Перегрузка операторов C++ как реализация поведения с предопределенным смыслом. Дружественность как механизм нарушения инкапсуляции. Достоинства и недостатки механизма дружественности. Статические поля и методы классов. Инициализация статических полей. 5.2. Реализация отношений между объектами и классами Ассоциация и агрегация объектов и классов. Зависимость по времени жизни. Использование и зависимость от интерфейсов. Объекты при передаче параметров и возврате из методов. Варианты реализации отношения клиент-сервер. Внутренние классы</p>
6	<p>6.1. Наследование как средство организации иерархий классов. Принцип замещения Лисковской. 6.2. Одиночное наследование. Понятие производного класса. Управление доступом в производных классах. Конструкторы и деструкторы, совмещение имен методов при наследовании, иерархии. Абстрактные классы и виртуальные функции. Виртуальный полиморфизм. Информация о типе на этапе выполнения. RTTI. 6.3. Множественное наследование Проблема множественного наследования. Виртуальное наследование как средство разрешения коллизий. Порядок вызовов конструкторов и деструкторов при множественном наследовании. Чистые виртуальные классы, понятие интерфейса. Принципы дизайна иерархий классов. OCP, DIP, ISP. 6.4. Пространства имен. 6.4.1. Пространства имен как средство реализации модульности. Поиск имен и разрешение конфликтов. Объединение пространств имен. Принципы дизайна пакетов. 6.5. Обработка исключений. Обработка ошибок. Группировка и перехват исключений. Управление ресурсами. Исключения и эффективность. Альтернативные методы обработки ошибок. Стандартные исключения.</p>
7	<p>7.1. Шаблоны классов. Определение шаблона. Инстанцирование. Параметры шаблонов и проверка типов.</p>

	7.2. Шаблоны функций. 7.3. Специализация. 7.4. Наследование и шаблоны.
8	8.1. Библиотека стандартных шаблонов Общие сведения (понятия контейнеров, итераторов и объектов-функций) Контейнеры (виды контейнеров, последовательные и ассоциативные контейнеры, адаптеры) Итераторы (итератор как обобщение указателя, классы итераторов). Алгоритмы (примеры алгоритмов с использованием итераторов: алгоритмы сортировки, алгоритмы, не изменяющие содержание контейнера, алгоритмы, изменяющие содержание контейнера) 8.2. Библиотека ввода-вывода Потоки вывода. Вывод типов определяемых пользователем. Потоки ввода. Ввод типов определяемых пользователем. Форматирование в потоках ввода-вывода. Буферизация.

5. Оценочные материалы для текущего и рубежного контроля успеваемости и промежуточной аттестации

Конечными результатами освоения программы дисциплины являются сформированные когнитивные дескрипторы «знать», «уметь», «владеть», расписанные по отдельным компетенциям. Формирование этих дескрипторов происходит в течение всего семестра по этапам в рамках различного вида занятий и самостоятельной работы.

В ходе изучения дисциплины предусматриваются *текущий, рубежный контроль и промежуточная аттестация*.

1.1. Оценочные материалы для текущего контроля. Цель текущего контроля – оценка результатов работы в семестре и обеспечение своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы обучающегося. Объектом текущего контроля являются конкретизированные результаты обучения (учебные достижения) по дисциплине

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины «Объектно-ориентированное программирование» и включает: ответы на теоретические вопросы на практическом занятии, решение практических задач и выполнение заданий на практическом занятии, самостоятельное выполнение индивидуальных домашних заданий (например, решение задач) с отчетом (защитой) в установленный срок.

Оценка качества подготовки на основании выполненных заданий ведется преподавателем (с обсуждением результатов), баллы начисляются в зависимости от сложности задания

5.1.1. Вопросы по темам дисциплины «Объектно-ориентированное программирование» (контролируемые компетенции ПКС-2):

Тема 1. Основные принципы объектно-ориентированного программирования

1. Первое поколение языков программирования.
2. Развитие алгоритмических абстракций, или второе поколение языков программирования.
3. Модуль как единица построения программных систем, третье поколение языков программирования.
4. Зарождение объектной модели, четвертое поколение языков программирования.
5. Объектные языки программирования, объектно-ориентированные языки программирования, объектно-ориентированный анализ, дизайн и проектирование.
6. Парадигмы программирования.
7. Абстрагирование. Инкапсуляция. Модульность. Иерархия. Типизация. Параллелизм. Сохраняемость.

Тема 2. Объектно-ориентированная модель

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

Тема 3. Классы

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

Тема 4. Основные алгоритмические отличия C++ от C

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

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

Тема 6. Средства объектно-ориентированного программирования C++

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

Тема 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.

Тема 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.
8. Вывод типов определяемых пользователем.
9. Потоки ввода.
10. Ввод типов определяемых пользователем.
11. Форматирование в потоках ввода-вывода.
12. Буферизация.

Критерии формирования оценок (оценивания) устного опроса

Устный опрос является одним из основных способов учёта знаний обучающегося по дисциплине «Объектно-ориентированное программирование». Развёрнутый ответ студента должен представлять собой связное, логически последовательное сообщение на заданную тему, показывать его умение применять определения и применять для решения практических задач.

В результате устного опроса знания, обучающегося оцениваются по следующей шкале:

3 балла, ставится, если обучающийся:

- 1) полно излагает изученный материал, даёт правильное определение понятий;
- 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только по учебнику, но и самостоятельно составленные;
- 3) излагает материал последовательно и правильно с точки зрения норм языка программирования.

2 балла, ставится, если обучающийся даёт ответ, удовлетворяющий тем же требованиям, что и для балла «3», но допускает 1-2 ошибки, которые сам же исправляет, и 1-2 недочёта в последовательности и языковом оформлении излагаемого.

1 балл, ставится, если обучающийся обнаруживает знание и понимание основных положений данной темы, но:

- 1) излагает материал неполно и допускает неточности в определении понятий;
- 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры;
- 3) излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.

0 баллов, ставится, если обучающийся обнаруживает незнание большей части соответствующего раздела изучаемого материала, допускает ошибки в определениях.

Баллы «3», «2», «1», могут ставиться не только за единовременный ответ, но и за рассредоточенный во времени, т.е. за сумму ответов, данных студентом на протяжении занятия.

5.1.2 Оценочные материалы для самостоятельной работы обучающегося (типовые задачи)(контролируемые компетенции ПКС-2)

Тема 1. Основные принципы объектно-ориентированного программирования

1. Класс «Студент». Описать пользовательский класс. Определить конструкторы, деструкторы, функции для просмотра и установки полей класса. Написать демонстрационную программу, в которой используются объекты пользовательского класса.
2. Класс «Служащий». Описать пользовательский класс. Определить конструкторы, деструкторы, функции для просмотра и установки полей класса. Написать демонстрационную программу, в которой используются объекты пользовательского класса.
3. Класс «Кадры». Описать пользовательский класс. Определить конструкторы, деструкторы, функции для просмотра и установки полей класса. Написать демонстрационную программу, в которой используются объекты пользовательского класса.
4. Класс «Библиотека». Описать пользовательский класс. Определить конструкторы, деструкторы, функции для просмотра и установки полей класса. Написать демонстрационную программу, в которой используются объекты пользовательского класса.
5. Класс «Экзамен». Описать пользовательский класс. Определить конструкторы, деструкторы, функции для просмотра и установки полей класса. Написать демонстрационную программу, в которой используются объекты пользовательского класса.
6. Класс «Персона». Описать пользовательский класс. Определить конструкторы, деструкторы, функции для просмотра и установки полей класса. Написать демонстрационную программу, в которой используются объекты пользовательского класса.
7. Класс «Автомобиль». Описать пользовательский класс. Определить конструкторы, деструкторы, функции для просмотра и установки полей класса. Написать демонстрационную программу, в которой используются объекты пользовательского класса.

8. Класс «Товар». Описать пользовательский класс. Определить конструкторы, деструкторы, функции для просмотра и установки полей класса. Написать демонстрационную программу, в которой используются объекты пользовательского класса.
9. Класс «Страна». Описать пользовательский класс. Определить конструкторы, деструкторы, функции для просмотра и установки полей класса. Написать демонстрационную программу, в которой используются объекты пользовательского класса.
10. Класс «Животное». Описать пользовательский класс. Определить конструкторы, деструкторы, функции для просмотра и установки полей класса. Написать демонстрационную программу, в которой используются объекты пользовательского класса.

Тема 2. Объектно-ориентированная модель

1. Студент, преподаватель, персона, заведующий кафедрой. Определить иерархию классов. Определить необходимые конструкторы и деструкторы. Определить статические компоненты. Предусмотреть метод добавления элемента в список. Просмотреть список вызовом виртуального метода.
2. Служащий, персона, рабочий, инженер. Определить иерархию классов. Определить необходимые конструкторы и деструкторы. Определить статические компоненты. Предусмотреть метод добавления элемента в список. Просмотреть список вызовом виртуального метода.
3. Рабочий, кадры, инженер, администрация. Определить иерархию классов. Определить необходимые конструкторы и деструкторы. Определить статические компоненты. Предусмотреть метод добавления элемента в список. Просмотреть список вызовом виртуального метода.
4. Деталь, механизм, изделие, узел. Определить иерархию классов. Определить необходимые конструкторы и деструкторы. Определить статические компоненты. Предусмотреть метод добавления элемента в список. Просмотреть список вызовом виртуального метода.
5. Организация, страховая компания, судостроительная компания, завод. Определить иерархию классов. Определить необходимые конструкторы и деструкторы. Определить статические компоненты. Предусмотреть метод добавления элемента в список. Просмотреть список вызовом виртуального метода.
6. Журнал, книга, печатное издание, учебник. Определить иерархию классов. Определить необходимые конструкторы и деструкторы. Определить статические компоненты. Предусмотреть метод добавления элемента в список. Просмотреть список вызовом виртуального метода.
7. Тест, экзамен, выпускной экзамен, испытание. Определить иерархию классов. Определить необходимые конструкторы и деструкторы. Определить статические компоненты. Предусмотреть метод добавления элемента в список. Просмотреть список вызовом виртуального метода.
8. Квитанция, накладная, документ, чек. Определить иерархию классов. Определить необходимые конструкторы и деструкторы. Определить статические компоненты. Предусмотреть метод добавления элемента в список. Просмотреть список вызовом виртуального метода.
9. Автомобиль, поезд, транспортное средство, экспресс. Определить иерархию классов. Определить необходимые конструкторы и деструкторы. Определить статические компоненты. Предусмотреть метод добавления элемента в список. Просмотреть список вызовом виртуального метода.
10. Млекопитающие, парнокопытные, птицы, животное. Определить иерархию классов. Определить необходимые конструкторы и деструкторы. Определить статические компоненты. Предусмотреть метод добавления элемента в список. Просмотреть список вызовом виртуального метода.

Тема 3. Классы

1. Студент, преподаватель, персона, заведующий кафедрой. Дополнить иерархию классов классами «группа». Написать для класса-группы метод-итератор. Написать функцию, которая выполняется для всех объектов группы. Написать демонстрационную программу, в которой создаются и разрушаются объекты-группы, а также демонстрируется использование итератора.
2. Служащий, персона, рабочий, инженер. Дополнить иерархию классов классами «группа». Написать для класса-группы метод-итератор. Написать функцию, которая выполняется для всех объектов группы. Написать демонстрационную программу, в которой создаются и разрушаются объекты-группы, а также демонстрируется использование итератора.
3. Рабочий, кадры, инженер, администрация. Дополнить иерархию классов классами «группа». Написать для класса-группы метод-итератор. Написать функцию, которая выполняется для всех объектов группы. Написать демонстрационную программу, в которой создаются и разрушаются объекты-группы, а также демонстрируется использование итератора.
4. Деталь, механизм, изделие, узел. Дополнить иерархию классов классами «группа». Написать для класса-группы метод-итератор. Написать функцию, которая выполняется для всех объектов группы. Написать демонстрационную программу, в которой создаются и разрушаются объекты-группы, а также демонстрируется использование итератора.
5. Организация, страховая компания, судостроительная компания, завод. Дополнить иерархию классов классами «группа». Написать для класса-группы метод-итератор. Написать функцию, которая выполняется для всех объектов группы. Написать демонстрационную программу, в которой создаются и разрушаются объекты-группы, а также демонстрируется использование итератора.
6. Журнал, книга, печатное издание, учебник. Дополнить иерархию классов классами «группа». Написать для класса-группы метод-итератор. Написать функцию, которая выполняется для всех объектов группы. Написать демонстрационную программу, в которой создаются и разрушаются объекты-группы, а также демонстрируется использование итератора.
7. Тест, экзамен, выпускной экзамен, испытание. Дополнить иерархию классов классами «группа». Написать для класса-группы метод-итератор. Написать функцию, которая выполняется для всех объектов группы. Написать демонстрационную программу, в которой создаются и разрушаются объекты-группы, а также демонстрируется использование итератора.
8. Квитанция, накладная, документ, чек. Дополнить иерархию классов классами «группа». Написать для класса-группы метод-итератор. Написать функцию, которая выполняется для всех объектов группы. Написать демонстрационную программу, в которой создаются и разрушаются объекты-группы, а также демонстрируется использование итератора.
9. Автомобиль, поезд, транспортное средство, экспресс. Дополнить иерархию классов классами «группа». Написать для класса-группы метод-итератор. Написать функцию, которая выполняется для всех объектов группы. Написать демонстрационную программу, в которой создаются и разрушаются объекты-группы, а также демонстрируется использование итератора.
10. Млекопитающие, парнокопытные, птицы, животное. Дополнить иерархию классов классами «группа». Написать для класса-группы метод-итератор. Написать функцию, которая выполняется для всех объектов группы. Написать демонстрационную программу, в которой создаются и разрушаются объекты-группы, а также демонстрируется использование итератора.

Тема 4. Основные алгоритмические отличия C++ от C

1. Вычислите длину окружности, площадь круга и объём шара одного и того же заданного радиуса.
2. Определите номер квадранта, в котором находится точка с заданными координатами (x, y).
3. Подсчитайте число и сумму положительных, число и произведение отрицательных элементов заданного массива $A(N)$.
4. Запишите подряд в массив $A(N)$ элементы заданного массива $B(2N)$, стоящие на чётных местах, а элементы, стоящие на нечётных местах, запишите в массив $C(N)$.
5. Дана матрица $A(N, M)$. Найдите её наибольший элемент и номера строки и столбца, на пересечении которых он находится.
6. Дана матрица $A(N, N)$ и целое P . Преобразуйте матрицу по правилу: строку с номером P сделайте столбцом с номером P , а столбец с номером P сделайте строкой с номером P .
7. Выясните, имеются ли в заданном векторе $A(N)$ два подряд идущих нулевых элемента.
8. Все элементы заданного вектора $A(N)$, начиная с первого по порядку положительного элемента, уменьшите на единицу.
9. В заданной целочисленной матрице $A(N, M)$ выведите на печать индексы первого положительного элемента, кратного заданному числу K . Если таких элементов в матрице нет, то выведите соответствующий текст. Элементы матриц просматривайте слева направо и сверху вниз.
10. Элементы заданной матрицы $A(N, N)$ переписывайте построчно в одномерный массив до тех пор, пока не встретится нулевой элемент.
11. В заданной матрице $A(N, M)$ найдите количество строк, не содержащих отрицательных чисел.
12. Дана матрица $A(N, M)$. Постройте вектор $B(N)$, элементы B_i которого равны единице, если элементы i -й строки образуют упорядоченную по убыванию или по возрастанию последовательность, и нулю во всех остальных случаях.

Тема 5. Средства объектного программирования языка C++

1. АТД – множество с элементами типа `char`. Дополнительно перегрузить следующие операции:
+ – добавить элемент в множество(типа `char + set`);
+ – объединение множеств;
= – проверка множеств на равенство.
2. АТД – однонаправленный список с элементами типа `char`. Дополнительно перегрузить следующие операции:
+ – объединить списки (`list+list`);
-- – удалить элемент из начала (типа `--list`);
= – проверка на равенство.
3. АТД – стек. Дополнительно перегрузить следующие операции:
+ – добавить элемент в стек;
-- – извлечь элемент из стека;
`bool()` – проверка, пустой ли стек.
4. АТД – очередь. Дополнительно перегрузить следующие операции:
+ – добавить элемент;
-- – извлечь элемент;
`bool()` – проверка, пустая ли очередь.
5. АТД – одномерный массив (вектор) вещественных чисел. Дополнительно перегрузить следующие операции:
+ – сложение векторов (`a[i]+b[i]` для всех i);

[] – доступ по индексу;
+ – добавить число к вектору (double+vector).

6. АТД – одномерный массив (вектор) вещественных чисел. Дополнительно перегрузить следующие операции:

- – вычитание векторов (a[i]-b[i] для всех i);

[] – доступ по индексу;

- – вычесть из вектора число (vector-double).

7. АТД – одномерный массив (вектор) вещественных чисел. Дополнительно перегрузить следующие операции:

* – умножение векторов (a[i]*b[i] для всех i);

[] – доступ по индексу;

* – умножить вектор на число (vector*double).

8. АТД – одномерный массив (вектор) вещественных чисел. Дополнительно перегрузить следующие операции:

int() – размер вектора;

() – установить новый размер;

- – вычесть из вектора число (vector-double);

[] – доступ по индексу;

9. АТД – двумерный массив (матрица) вещественных чисел. Дополнительно перегрузить следующие операции:

() – доступ по индексу;

* – умножение матриц;

* – умножение матрицы на число;

* – умножение числа на матрицу.

10. АТД – двумерный массив (матрица) вещественных чисел. Дополнительно перегрузить следующие операции:

() – доступ по индексу;

- – разность матриц;

- – вычесть из матрицы число;

= – проверка матриц на равенство.

Тема 6. Средства объектно-ориентированного программирования C++

1. Шаблон класса – одномерный массив. Дополнительно перегрузить следующие операции:

* – умножение массивов;

[] – доступ по индексу.

2. Шаблон класса – множество set. Дополнительно перегрузить следующие операции:

+ – добавить элемент в множество (типа set+item);

+ – объединение множеств;

* – пересечение множеств;

3. Шаблон класса – множество set. Дополнительно перегрузить следующие операции:

+ – добавить элемент в множество (типа item + set);

+ – объединение множеств;

= – проверка множеств на равенство. 64

4. Шаблон класса – однонаправленный список list. Дополнительно перегрузить следующие операции:

+ – добавить элемент в начало (list+item);
-- – удалить элемент из начала (--list);
= – проверка на равенство.

5. Шаблон класса – однонаправленный список list. Дополнительно перегрузить следующие операции:

+ – добавить элемент в начало (item+list);
-- – удалить элемент из начала (--list);
!= – проверка на неравенство.

6. Шаблон класса – однонаправленный список list. Дополнительно перегрузить следующие операции:

+ – добавить элемент в конец (list+item);
-- – удалить элемент из конца (типа list--);
!= – проверка на неравенство.

7. Шаблон класса – стек stack. Дополнительно перегрузить следующие операции:

+ – добавить элемент в стек;
-- – извлечь элемент из стека;
bool() – проверка, пустой ли стек.

8. Шаблон класса – очередь queue. Дополнительно перегрузить следующие операции:

+ – добавить элемент;
-- – извлечь элемент;
bool() – проверка, пустая ли очередь.

9. Шаблон класса – одномерный массив. Дополнительно перегрузить следующие операции:

+ – сложение массивов;
[] – доступ по индексу;
+ – сложить элемент с массивом.

10. Шаблон класса – одномерный массив. Дополнительно перегрузить следующие операции:

- – вычитание массивов;
[] – доступ по индексу;
- – вычесть из массива элемент.

Тема 7.Обобщенное программирование

1. Класс - контейнер ВЕКТОР с элементами типа int. Реализовать операции:

[] – доступа по индексу;
() – определение размера вектора;
+ число – добавляет константу ко всем элементам вектора;

Пользовательский класс Time для работы с временными интервалами. Интервал должен быть представлен в виде двух полей: минуты типа int и секунды типа int. при выводе минуты отделяются от секунд двоеточием.

2. Класс- контейнер ВЕКТОР с элементами типа int. Реализовать операции:

[] – доступа по индексу;
int() – определение размера вектора;
+ вектор –сложение элементов векторов a[i]+b[i];

Пользовательский класс Time для работы с временными интервалами. Интервал должен быть представлен в виде двух полей: минуты типа int и секунды типа int. при выводе минуты отделяются от секунд двоеточием.

3.Класс- контейнер ВЕКТОР с элементами типа int. Реализовать операции:

[] – доступа по индексу;

+ вектор – сложение элементов векторов a[i]+b[i];

+ число – добавляет константу ко всем элементам вектора;

Пользовательский класс Time для работы с временными интервалами. Интервал должен быть представлен в виде двух полей: минуты типа int и секунды типа int. при выводе минуты отделяются от секунд двоеточием.

4. Класс- контейнер МНОЖЕСТВО с элементами типа int. Реализовать операции:

[] – доступа по индексу;

() – определение размера множества;

+ – объединение множеств;

Пользовательский класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: типа long для рублей и типа int для копеек. Дробная часть числа при выводе на экран должна быть отделена от целой части запятой.

5. Класс- контейнер МНОЖЕСТВО с элементами типа int. Реализовать операции:

[] – доступа по индексу;

int() – определение размера вектора;

* – пересечение множеств;

Пользовательский класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: типа long для рублей и типа int для копеек. Дробная часть числа при выводе на экран должна быть отделена от целой части запятой.

6. Класс- контейнер МНОЖЕСТВО с элементами типа int. Реализовать операции:

[] – доступа по индексу;

== - проверка на равенство;

> число – принадлежность числа множеству;

Пользовательский класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: типа long для рублей и типа int для копеек. Дробная часть числа при выводе на экран должна быть отделена от целой части запятой.

7. Класс- контейнер СПИСОК с ключевыми значениями типа int. Реализовать операции:

[] – доступа по индексу;

int() – определение размера списка;

+ вектор – сложение элементов списков a[i]+b[i];

Пользовательский класс Money для работы с денежными суммами. Число должно быть представлено двумя полями: типа long для рублей и типа int для копеек. Дробная часть числа при выводе на экран должна быть отделена от целой части запятой.

8.Класс- контейнер СПИСОК с ключевыми значениями типа int. Реализовать операции:

[] – доступа по индексу;

() – определение размера вектора;

+ число – добавляет константу ко всем элементам вектора;

Пользовательский класс Pair (пара чисел). Пара должна быть представлено двумя полями: типа int для первого числа и типа double для второго. Первое число при выводе на экран должно быть отделено от второго числа двоеточием.

9. Класс- контейнер СПИСОК с ключевыми значениями типа int. Реализовать операции:

[] – доступа по индексу;

- + вектор – сложение элементов списков $a[i]+b[i]$;
- + число – добавляет константу ко всем элементам списка;

Пользовательский класс `Pair` (пара чисел). Пара должна быть представлена двумя полями: типа `int` для первого числа и типа `double` для второго. Первое число при выводе на экран должно быть отделено от второго числа двоеточием.

10. Класс- контейнер СПИСОК с ключевыми значениями типа `int`. Реализовать операции:

- `[]` – доступа по индексу;
- `()` – определение размера списка;
- `*` число – умножает все элементы списка на число;

Пользовательский класс `Pair` (пара чисел). Пара должна быть представлена двумя полями: типа `int` для первого числа и типа `double` для второго. Первое число при выводе на экран должно быть отделено от второго числа двоеточием.

Тема 8. Стандартная библиотека C++

1. `Vector`. Определение пользовательского класса. Определение компонентных функций для работы с контейнером. Объяснение этих функций и алгоритмов STL.
2. `List`. Определение пользовательского класса. Определение компонентных функций для работы с контейнером. Объяснение этих функций и алгоритмов STL.
3. `Deque`. Определение пользовательского класса. Определение компонентных функций для работы с контейнером. Объяснение этих функций и алгоритмов STL.
4. `Stack`. Определение пользовательского класса. Определение компонентных функций для работы с контейнером. Объяснение этих функций и алгоритмов STL.
5. `Queue`. Определение пользовательского класса. Определение компонентных функций для работы с контейнером. Объяснение этих функций и алгоритмов STL.
6. `Map`. Определение пользовательского класса. Определение компонентных функций для работы с контейнером. Объяснение этих функций и алгоритмов STL.
7. `Multimap`. Определение пользовательского класса. Определение компонентных функций для работы с контейнером. Объяснение этих функций и алгоритмов STL.
8. `Set`. Определение пользовательского класса. Определение компонентных функций для работы с контейнером. Объяснение этих функций и алгоритмов STL.
9. `Multiset`. Определение пользовательского класса. Определение компонентных функций для работы с контейнером. Объяснение этих функций и алгоритмов STL.
10. `String`. Определение пользовательского класса. Определение компонентных функций для работы с контейнером. Объяснение этих функций и алгоритмов STL.

Методические рекомендации по решению задач.

При решении задач необходимо внимательно ознакомиться с контентом по соответствующему вопросу темы. Разобрать основные понятия темы, разобрать примеры из лабораторных работ. Кратко описать алгоритм решения задачи, объяснить критерии выбора соответствующей структуры данных. Выполнить кодирование задачи на языке программирования. Выполнить отладку и тестирование программы. В отчет включить листинг программы, полный набор тестов и результаты работы программы на этих тестах.

Критерии формирования оценок по заданиям для самостоятельной работы студента (типовые задачи):

«отлично» (5 баллов) - обучающийся показал глубокие знания материала по поставленным вопросам, грамотно, логично его излагает, структурировал и детализировал информацию, избегая простого повторения информации из текста, информация представлена в переработанном виде. Свободно использует необходимые формулы при решении задач;

«хорошо» (4 балл) - обучающийся твердо знает материал, грамотно его излагает, не допускает существенных неточностей в процессе решения задач;

«удовлетворительно» (3 балла) - обучающийся имеет знания основного материала по поставленным вопросам, но не усвоил его деталей, допускает отдельные неточности при решении

задач;

«неудовлетворительно» (менее 3 баллов) – обучающийся допускает грубые ошибки в ответе на поставленные вопросы и при решении задач

5.2. Оценочные материалы для рубежного контроля. Рубежный контроль осуществляется по более или менее самостоятельным разделам – учебным модулям курса и проводится по окончании изучения материала модуля в заранее установленное время. Рубежный контроль проводится с целью определения качества усвоения материала учебного модуля в целом. В течение семестра проводится **три таких контрольных мероприятия по графику**. В качестве форм рубежного контроля можно использовать тестирование (письменное или компьютерное), проведение коллоквиума или контрольных работ. Выполняемые работы должны храниться на кафедре течения учебного года и по требованию предоставляться в Управление контроля качества. На рубежные контрольные мероприятия рекомендуется выносить весь программный материал (все разделы) по дисциплине.

5.2.1. Оценочные материалы для контрольной работы: контролируемые компетенции, ПКС-2):

Типовые Варианты контрольных работ:

Вариант 1.

Задание 1. Общая структура программы на языке Паскаль.

Задание 2. Вычислить длину окружности, площадь круга и объем шара заданного радиуса.

Задание 3. Проверить, можно ли построить прямоугольный треугольник из отрезков с длинами a , b , c .

Вариант 2.

Задание 1. Операторы цикла в языке Паскаль.

Задание 2. В заданном массиве $A(N)$ подсчитать сумму элементов, кратных 5.

Задание 3. В заданной матрице $A(N, M)$ подсчитать количество строк, состоящих только из четных элементов.

Вариант 3.

Задание 1. Процедуры и функции обработки строк.

Задание 2. В заданном тексте найти самое длинное симметричное слово.

Задание 3. Упорядочить слова заданного текста в алфавитном порядке.

Вариант 4.

Задание 1. Процедуры и функции обработки файлов.

Задание 2. Найти максимальный компонент заданного файла.

Задание 3. Объединить два заданных файла и записать в третий файл.

Вариант 5.

Задание 1. Структура модуля.

Задание 2. Разработать текст модуля для решения геометрических задач.

Задание 3. Составить программу решение задачи с использованием модуля.

Вариант 6.

Задание 1. Динамические структуры данных.

Задание 2. Создать стек для хранения натуральных чисел.

Задание 3. Создать дерево двоичного поиска из заданного списка вещественных чисел.

Вариант 7.

Задание 1. Рекурсия.

Задание 2. Составить рекурсивную и не рекурсивную функцию вычисления n -го элемента ряда Фибоначчи и сравнить их время работы для различных входных данных.

Задание 3. В заданном графе найти самый короткий путь между двумя вершинами.

Критерии формирования оценок по контрольным точкам (контрольные работы; коллоквиум)

(5 баллов) - ставится за работу, выполненную полностью без ошибок и недочетов; обучающийся демонстрирует знание теоретического и практического материала по теме практической работы, решено 100% задач;

(4 баллов) – ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов. Обучающийся демонстрирует знание теоретического и практического материала по теме практической работы, допуская незначительные неточности при решении задач, решено 70% задач;

(3 балла) – ставится за работу, если бакалавр правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой. Обучающийся затрудняется с правильной оценкой предложенной задачи, дает неполный ответ, решено 55% задач

(менее 3 баллов) – ставится за работу, если число ошибок и недочетов превысило норму для оценки 3 или правильно выполнено менее 2/3 всей работы. Обучающийся дает неверную оценку ситуации, решено менее 50 % задач.

5.2.2. Оценочные материалы: Типовые тестовые задания по дисциплине
«Объектно-ориентированное программирование»
(контролируемые компетенции ПКС-2)

I:

S: Для чего нужны классы?

- +: для определения новых типов в программе;
- : для упрощения работы с функциями;
- +: для соединения данных и операций над ними;
- : для упрощения работы с указателями.

I:

S: Методы класса определяют:

- +: какие операции можно выполнять с объектами данного класса;
- : какие типы значений могут принимать данные-члены класса;
- : каким образом реализуется защита данных-членов класса.

I:

S: Атрибуты (данные-члены) класса могут быть:

- : только целыми числами;
- : любыми встроенными типами;
- +: любого определенного в программе типа.

I:

S:

Какая функция выполняет начальную инициализацию данных в классе?

- +: конструктор
- : геттер
- : сеттер
- : деструктор

I:

S:

Отметьте правильные утверждения

- +: конструкторы класса не наследуются
- : конструкторов класса может быть несколько, их синтаксис определяется программистом
- +: конструкторов класса может быть несколько, но их синтаксис должен подчиняться правилам перегрузки функций
- : конструктор возвращает указатель на объект
- +: конструктор не возвращает значение

I:

S: Что такое деструктор?

- : Деструктор - это функция, которая должна открывать динамическую область для экземпляра класса
- +: Деструктор - это специальная функция-элемент, которая должна уничтожить экземпляр класса после завершения его работы
- : Деструктор - это специальная функция-элемент, которая должна отслеживать данные в экземпляре класса в процессе работы

I:

S:

Какого спецификатора доступа в классах нет?

- : public
- : private
- : protected
- +: hidden

I:

S:

Укажите корректное объявление класса!

- +: class A { int x; };
- : class B { }
- : object A { int x; };
- : public class A { }

I:

S: Какая из записей соответствует обращению к атрибуту arg класса A в определении метода этого же класса?

- +: this → arg;
- +: arg;
- : A → arg;
- : A → this → arg.

I:

S: Если имеется код class A { public: int a; }; A *obj; то как обратиться к переменной a?

Варианты ответа:

- : obj.a;
- +: (*obj) → a;
- +: obj → a;
- : obj :: a.

I:

S:

Если имеется код class A { public: int a, b, c; }; A *obj; то как обратиться к переменной b?

Варианты ответа:

- : (*obj) → b;
- : A :: b;

+: 3) (*obj).b;
-: obj->a.b.

I:

S: Каков будет результат выполнения следующего кода:

```
class A {  
    public:  
        int inc (int x) {return x++;}  
        int inc (short x) (return x+2;}  
};  
A obj; int y=5; cout << obj.inc(y); ?  
+: 5  
-: 6  
-: 7  
-: ошибка при компиляции.
```

I:

S: Каков будет результат выполнения следующего кода:

```
class A {  
    public: int y;  
        int inc (int x) {return y++;}  
        int inc (short x) {return x+y;}  
};  
A obj; int y=5; obj.y= 6; cout << obj.inc(y); ?  
-: 5  
+: 6  
-: 11  
-: 7  
-: ошибка при компиляции.
```

I:

S: Какими по умолчанию объявляются элементы структуры?

+: private;
+: public;
-: protected;
-: по умолчанию не объявляются.

5.2.3. Оценочные материалы для промежуточной аттестации.

Целью промежуточных аттестаций по дисциплине является оценка качества освоения дисциплины обучающимися.

Промежуточная аттестация предназначена для объективного подтверждения и оценивания достигнутых результатов обучения после завершения изучения дисциплины. Осуществляется в конце семестра и представляет собой итоговую оценку знаний по дисциплине «Объектно-ориентированное программирование» в виде проведения зачета.

Промежуточная аттестация может проводиться в устной, письменной форме, и в форме тестирования. На промежуточную аттестацию отводится до 30 баллов.

ВОПРОСЫ НА ЗАЧЕТ (контролируемые компетенции ПКС-2)

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. Мета модель. Инстанцирование.
12. Классы. Структура класса. Абстрактные классы и интерфейсы.
13. Классы. Отношения между классами. Ассоциация и агрегация.
14. Классы. Иерархии классов. Зависимость.
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, using директива.
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.
25. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании.
26. Варианты реализации отношения клиент-сервер. Объекты при передаче параметров и возврате из методов.
27. Исключения в C++. Обработка исключений.
28. Шаблоны классов и шаблоны функций. Специализация.
29. Основы STL. Структура и назначение. Контейнеры. Алгоритмы
30. Стандартная библиотека, ввод-вывода.

Критерии формирования оценок по промежуточной аттестации:

«зачтено» – получают обучающиеся, которые относительно полно ориентируются в материале, отвечают без затруднений, допускают незначительное количество ошибок. Обучающийся способен к выполнению сложных заданий. Работа выполнена полностью, но имеются не более одной негрубой ошибки и одного недочета, не более трех недочетов. Допускаются незначительные неточности при решении задач, решено не менее 60% задач;

«незачтено» – получают обучающиеся, у которых недостаточно высок уровень владения материалом. В процессе ответа допускаются грубые ошибки и затруднения при изложении материала. Обучающийся правильно выполнил менее 2/3 всей работы. Обучающийся затрудняется с правильной оценкой предложенной задачи, дает неполный ответ, решено менее 60% задач;

6. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Максимальная сумма (100 баллов), набираемая студентом по дисциплине включает две составляющие:

– *первая составляющая* – оценка регулярности, своевременности и качества выполнения студентом учебной работы по изучению дисциплины в течение периода изучения дисциплины (семестра, или нескольких семестров) (сумма – не более 70 баллов). Баллы, характеризующие успеваемость студента по дисциплине, набираются им в течение всего периода обучения за изучение отдельных тем и выполнение отдельных видов работ.

– *вторая составляющая* – оценка знаний студента по результатам промежуточной аттестации (не более 30 –баллов).

Критерием оценки уровня сформированности компетенций в рамках учебной дисциплины «Объектно-ориентированное программирование» в VIII семестре является зачет.

Общий балл текущего и рубежного контроля складывается из следующих составляющих приложение 2.

Целью промежуточных аттестаций по дисциплине является оценка качества освоения дисциплины обучающимися.

Критерии оценки качества освоения дисциплины (Приложение)

Оценка «зачтено» – от 61 до 100 баллов – теоретическое содержание курса освоено, необходимые практические навыки работы сформированы, выполненные учебные задания содержат незначительные ошибки. На зачете студент демонстрирует твердое знание основного (программного) материала, умеет четко, грамотно, без существенных неточностей отвечать на поставленные вопросы.

Оценка «не зачтено» – от 36 до 60 баллов – теоретическое содержание курса освоено не полностью, необходимые практические навыки работы сформированы частично, выполненные учебные задания содержат грубые ошибки. На зачете студент демонстрирует знание только основного материала, ответы содержат неточности, слабо аргументированы, нарушена последовательность изложения материала

Типовые задания, обеспечивающие формирование компетенций ПКС-2 представлены в таблице 7

Таблица 7. Результаты освоения учебной дисциплины, подлежащие проверке

<i>Результаты обучения (компетенции)</i>	<i>Индикаторы достижения компетенции</i>	<i>Основные показатели оценки результатов обучения</i>	<i>Вид оценочного материала, обеспечивающего формирование компетенций</i>
ПКС-2. Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных информационных технологий, программирования и компьютерной техники	ПКС-2.1. Способен использовать основные методы проектирования и производства программного продукта и программных комплексов, их сопровождения, администрирования и развития (эволюции)	<p>ПКС-2.1. З-1. Знает арсенал и области применения современных научных методов и информационных технологий, необходимых для решения задач, имеющих естественно-научное содержание и возникающих при выполнении профессиональных функций</p> <p>ПКС-2.1. У-1. Умеет описывать проблемы и ситуации профессиональной деятельности на основе знаний математического аппарата и естественнонаучных дисциплин и формулировать задачу профессиональной деятельности в области прикладной математики и информатики аппарата и естественнонаучных дисциплин</p> <p>ПКС-2.1. В-1. Владеет навыками производить статистические расчеты с применением соответствующих математических методов и информационных технологий, а также проводить</p>	<p>Типовые оценочные материалы для устного опроса (раздел 5.1.1)</p> <p>Типовые тестовые задания (раздел 5.2.2.);</p> <p>Оценочные материалы для самостоятельной работы (типовые задачи раздел 5.1.2.);</p> <p>Оценочные материалы для контрольной работы (раздел 5.2.1)</p> <p>Оценочные материалы для промежуточной</p>

		<p>последующую аналитическую работу с полученными данными</p> <p>ПКС-2.2. 3-1. Знает методологии разработки программного обеспечения и технологии программирования</p> <p>ПКС-2.2. У-1. Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов</p> <p>ПКС-2.2. В-1. Владеет навыками программирования элементов компьютерной графики и навыками создания правильных, геометрических и реалистичных изображений на экране компьютера</p>	<p>аттестации (<i>раздел 5.2.3</i>)</p>
--	--	--	---

Таким образом, выполнение типовых заданий, представленных в разделе 5 «Оценочные материалы для текущего и рубежного контроля успеваемости и промежуточной аттестации» позволит обеспечить:

способность применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности (ПКС-2)

7. Учебно-методическое обеспечение дисциплины

7.1. Нормативно-законодательные акты

1. Приказ Минобрнауки России от 06.04.2021 № 245 "Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры" (Зарегистрировано в Минюсте России 13.08.2021 N 64644).
2. Федеральный государственный образовательный стандарт по образовательным программам ВО (ФГОС 3++) по направлению подготовки 01.03.02 Прикладная математика и информатика (уровень бакалавриата). Приказ Министерства образования и науки Российской Федерации от 10 января 2018г. №9 (Зарегистрировано в министерстве юстиции Российской Федерации 06 февраля 2018г. № 49937);
3. Федеральный закон "Об образовании в Российской Федерации" от 29.12.2012 N 273-ФЗ http://www.consultant.ru/document/cons_doc_LAW_140174/
4. Программа «Цифровая экономика», утвержденная распоряжением Правительства Российской Федерации от 28 июля 2017 г. №1632-р.
5. Указ Президента Российской Федерации от 9 мая 2017 г. №203 «О Стратегии развития информационного общества в Российской Федерации на 2017–2030 годы».

7.2. Основная литература

	Название	Режим доступа
1.	Мейер Б. Основы объектно-ориентированного проектирования— 2-е изд. М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 765 с. — 2227-8397.	iprbookshop.ru/73692
2.	Васильев А.Н. Объектно-ориентированное программирование на С++ — СПб. : Наука и Техника, 2016. — 544 с. — 978-5-94387-984-5.	iprbookshop.ru/60648
3.	Букунов С.В. Основы объектно-ориентированного программирования: учебное пособие — СПб. : Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2017. — 196 с. — 978-5-9227-0713-8.	iprbookshop.ru/74339
4.	Комлев Н.Ю. Объектно Ориентированное Программирование. Хорошая книга для Хороших Людей. — М. : СОЛОН-ПРЕСС, 2014. — 298 с. — 978-5-91359-138-8.	iprbookshop.ru/26923

7.3 Дополнительная литература

	Название	Режим доступа
1.	Мейер Б. Объектно-ориентированное программирование и программная инженерия. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 285 с. — 2227-8397.	iprbookshop.ru/39552
2.	Николаев Е.И. Объектно-ориентированное программирование: учебное пособие. — Ставрополь: Северо-Кавказский федеральный университет, 2015. — 225 с. — 2227-8397.	iprbookshop.ru/62967
3.	Николаев Е.И. Объектно-ориентированное программирование. Часть 1: лабораторный практикум. — Ставрополь: Северо-Кавказский федеральный университет, 2015. — 183 с. — 2227-8397.	iprbookshop.ru/62966
4.	Николаев Е.И. Объектно-ориентированное программирование. Часть 2: лабораторный практикум. — Ставрополь: Северо-Кавказский федеральный университет, 2015. — 156 с. — 2227-8397.	iprbookshop.ru/63218
5.	Фридман А.Л. Язык программирования Си++. 2-е изд. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 218 с. — 5-9556-0017-5.	iprbookshop.ru/73738
6.	Русанова Я.М. С++ как второй язык в обучении приемам и технологиям программирования. — Ростов-на-Дону: Южный федеральный университет, 2010. — 200 с. — 978-5-9275-0749-8.	iprbookshop.ru/47120
7.	Иванов В.Б. Прикладное программирование на С/С++. С нуля до мультимедийных и сетевых приложений. — М. : СОЛОН-ПРЕСС, 2011. — 240 с. — 5-98003-279-7	iprbookshop.ru/65139

7.4. Интернет-ресурсы

1. <http://www.cyberforum.ru/> - Форум программистов и сисадминов «Киберфорум»
2. <http://intuit.ru> – Национальный Открытый Университет «Интуит»
3. <http://www.osp.ru> – Издательство «Открытые системы»
4. <http://www.cnews.ru> – Издание о высоких технологиях
5. <http://habrahabr.ru/> - Сообщество IT-профессионалов

Перечень актуальных электронных информационных баз данных, к которым обеспечен доступ пользователям КБГУ (2022-2023 уч. год)

№	Наименование электрон-	Краткая характеристика	Адрес сайта	Наименование орга-	Условия доступа
---	------------------------	------------------------	-------------	--------------------	-----------------

п/п	ного ресурса			низации- владельца; реквизиты договора	
1.	Научная электронная библиотека (НЭБ РФФИ)	Электр. библиотека научных публикаций - около 4000 иностранных и 3900 отечественных научных журналов, рефераты публикаций 20 тыс. журналов, а также описания 1,5 млн. зарубежных и российских диссертаций; 2800 росс. журналов на безвозмездной основе	http://elibrary.ru	ООО «НЭБ»	Полный доступ
2.	База данных Science Index (РИНЦ)	Национальная информационно-аналитическая система, аккумулирующая более 6 миллионов публикаций российских авторов, а также информацию об их цитировании из более 4500 российских журналов.	http://elibrary.ru	ООО «НЭБ» Лицензионный договор Science Index №SIO-741/2022 от 19.07.2022 Активен до 31.07.2023г.	Авторизованный доступ. Позволяет дополнять и уточнять сведения о публикациях ученых КБГУ, имеющих в РИНЦ
3.	ЭБС «Консультант студента»	13800 изданий по всем областям знаний, включает более чем 12000 учебников и учебных пособий для ВО и СПО, 864 наименований журналов и 917 монографий.	http://www.studmedlib.ru http://www.medcollegelibrary.ru	ООО «Консультант студента» (г. Москва) Договор №750КС/07-2022 От 26.09.2022 Активен до 30.09.2023г.	Полный доступ (регистрация по IP-адресам КБГУ)
4.	«Электронная библиотека технического вуза» (ЭБС «Консультант студента»)	Коллекция «Медицина (ВО) ГЭОТАР-Медиа. Books in English (книги на английском языке)»	http://www.studmedlib.ru	ООО «Политехресурс» (г. Москва) Договор №701КС/02-2022 от 13.04.2022 Активен до	Полный доступ (регистрация по IP-адресам КБГУ)

				вен до 19.04.2023г.	
5.	ЭБС «Лань»	Электронные версии книг ведущих издательств учебной и научной литературы (в том числе университетских издательств), так и электронные версии периодических изданий по различным областям знаний.	https://e.lanbook.com/	ООО «ЭБС ЛАНЬ» (г. Санкт-Петербург) Договор №6ЕП/223 от 15.02.2022 Активен до 28.02.2023г.	Полный доступ (регистрация по IP-адресам КБГУ)
6.	Национальная электронная библиотека РГБ	Объединенный электронный каталог фондов российских библиотек, содержащий 4 331 542 электронных документов образовательного и научного характера по различным отраслям знаний	https://nab.ru/	ФГБУ «Российская государственная библиотека» Договор №101/НЭБ/166 6-п от 10.09.2020г. Сроком на 5 лет	Доступ с электронного читального зала библиотеки КБГУ
7.	ЭБС «IPRbooks»	107831 публикаций, в т.ч.: 19071 – учебных изданий, 6746 – научных изданий, 700 коллекций, 343 журнала ВАК, 2085 аудио изданий.	http://iprbooks.ru/	ООО «Ай Пи Эр Медиа» (г. Саратов) Договор №9200/22П от 08.04.2022 Активен до 02.04.2023г.	Полный доступ (регистрация по IP-адресам КБГУ)
8.	ЭБС «Юрайт» для СПО	Электронные версии учебной и научной литературы издательств «Юрайт» для СПО и электронные версии периодических изданий по различным областям знаний.	https://www.biblio-online.ru/	ООО «Электронное издательство ЮРАЙТ» (г. Москва) Договор №192/ЕП-223 От 29.10.2021 Активен до 31.10.2022 г.	Полный доступ (регистрация по IP-адресам КБГУ)
9.	Polpred.com. Новости. Обзор СМИ. Россия и зарубежье	Обзор СМИ России и зарубежья. Полные тексты + аналитика из 600 изданий по 53 отраслям	http://polpred.com	ООО «Полпред справочники» Безвозмездно (без официального	Доступ по IP-адресам КБГУ

				договора)	
10.	Президентская библиотека им. Б.Н. Ельцина	Более 500 000 электронных документов по истории Отечества, российской государственности, русскому языку и праву	http://www.prilib.ru	ФГБУ «Президентская библиотека им. Б.Н. Ельцина» (г. Санкт-Петербург) Соглашение от 15.11.2016г. Бессрочный	Авторизованный доступ из библиотеки (ауд. №115, 214)

7.4 Методические указания по проведению различных учебных занятий, к курсовому проектированию и другим видам самостоятельной работы.

Учебная работа по дисциплине «Объектно-ориентированное программирование» состоит из контактной работы (лекции, практические занятия) и самостоятельной работы. Доля контактной учебной работы в общем объеме времени, отведенном для изучения дисциплины, составляет 48,2 % (в том числе лекционных занятий – 21,9%, практических занятий – 26,3%), доля самостоятельной работы – 38 %. Соотношение лекционных, семинарских, лабораторных и практических занятий к общему количеству часов соответствует учебному плану Направления 01.03.02 – Прикладная математика и информатика.

Для подготовки к практическим занятиям необходимо рассмотреть контрольные вопросы, при необходимости обратиться к рекомендуемой литературе, записать непонятные моменты в вопросах для уяснения их на предстоящем занятии.

Методические рекомендации по изучению дисциплины для обучающихся

Цель курса «Объектно-ориентированное программирование» - ознакомление студентов с основными понятиями, структурами, методами, алгоритмами, парадигмами современного программирования и приобретение навыков решения задач с использованием современных языков и технологий программирования.

Приступая к изучению дисциплины, обучающемуся необходимо ознакомиться с тематическим планом занятий, списком рекомендованной учебной литературы. Следует уяснить последовательность выполнения индивидуальных учебных заданий, занести в свою рабочую тетрадь темы и сроки проведения семинаров, написания учебных и творческих работ. При изучении дисциплины обучающиеся выполняют следующие задания: изучают рекомендованную учебную и научную литературу; пишут контрольные работы, готовят доклады и сообщения к практическим занятиям; выполняют самостоятельные творческие работы, участвуют в выполнении практических заданий. Уровень и глубина усвоения дисциплины зависят от активной и систематической работы на лекциях, изучения рекомендованной литературы, выполнения контрольных письменных заданий

Курс изучается на лекциях, семинарах, при самостоятельной и индивидуальной работе обучающихся. Обучающийся для полного освоения материала должен не пропускать занятия и активно участвовать в учебном процессе. Лекции включают все темы и основные вопросы теории и практики страхования. Для максимальной эффективности изучения необходимо постоянно вести конспект лекций, знать рекомендуемую преподавателем литературу, позволяющую дополнить знания и лучше подготовиться к семинарским занятиям.

В соответствии с учебным планом на каждую тему выделено необходимое количество часов практических занятий, которые проводятся в соответствии с вопросами, рекомендованными к изучению по определенным темам. Обучающиеся должны регулярно готовиться к семинарским занятиям и участвовать в обсуждении вопросов. При подготовке к занятиям следует руководствоваться конспектом лекций и рекомендованной литературой. Тематический план дисциплины

ны, учебно-методические материалы, а также список рекомендованной литературы приведены в рабочей программе

В ходе изучения дисциплины обучающийся имеет возможность подготовить реферат по выбранной из предложенного в Рабочей программе списка теме. Выступление с докладом по реферату в группе проводится в форме презентации с использованием мультимедийной техники.

Методические рекомендации при работе над конспектом во время проведения лекции

В процессе лекционных занятий целесообразно конспектировать учебный материал. Для этого используются общие и утвердившиеся в практике правила, и приемы конспектирования лекций:

Конспектирование лекций ведется в специально отведенной для этого тетради, каждый лист которой должен иметь поля, на которых делаются пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

Целесообразно записывать тему и план лекций, рекомендуемую литературу к теме. Записи разделов лекции должны иметь заголовки, подзаголовки, красные строки. Для выделения разделов, выводов, определений, основных идей можно использовать цветные карандаши и фломастеры.

Названные в лекции ссылки на первоисточники надо пометить на полях, чтобы при самостоятельной работе найти и вписать их. В конспекте дословно записываются определения понятий, категорий и законов. Остальное должно быть записано своими словами.

Каждому студенту необходимо выработать и использовать допустимые сокращения наиболее распространенных терминов и понятий.

Методические рекомендации по подготовке к практическим занятиям

Практические (семинарские) занятия – составная часть учебного процесса, групповая форма занятий при активном участии студентов. Практические занятия способствуют углубленному изучению наиболее сложных проблем науки и служат основной формой подведения итогов самостоятельной работы обучающихся. Целью практических занятий является углубление и закрепление теоретических знаний, полученных обучающимися на лекциях и в процессе самостоятельного изучения учебного материала, а, следовательно, формирование у них определенных умений и навыков.

В ходе подготовки к семинарскому занятию необходимо прочитать конспект лекции, изучить основную литературу, ознакомиться с дополнительной литературой, выполнить выданные преподавателем практические задания. При этом учесть рекомендации преподавателя и требования программы. Дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы.

Желательно при подготовке к практическим занятиям по дисциплине одновременно использовать несколько источников, раскрывающих заданные вопросы.

На практических занятиях обучающиеся учатся грамотно излагать проблемы, свободно высказывать свои мысли и суждения, рассматривают ситуации, способствующие развитию профессиональной компетентности. Следует иметь в виду, что подготовка к практическому занятию зависит от формы, места проведения семинара, конкретных заданий и поручений. Это может быть написание доклада, эссе, реферата (с последующим их обсуждением), коллоквиум.

Методические рекомендации по организации самостоятельной работы

Самостоятельная работа (по В.И. Далю «самостоятельный – человек, имеющий свои твердые убеждения») осуществляется при всех формах обучения: очной и заочной.

Самостоятельная работа обучающихся - способ активного, целенаправленного приобретения студентом новых для него знаний и умений без непосредственного участия в этом процесса преподавателей. Повышение роли самостоятельной работы обучающихся при проведении различных видов учебных занятий предполагает:

– оптимизацию методов обучения, внедрение в учебный процесс новых технологий обучения, повышающих производительность труда преподавателя, активное использование информационных технологий, позволяющих обучающемуся в удобное для него время осваивать учебный материал;

- широкое внедрение компьютеризированного тестирования;
- совершенствование методики проведения практик и научно-исследовательской работы обучающихся, поскольку именно эти виды учебной работы в первую очередь готовят обучающихся к самостоятельному выполнению профессиональных задач;
- модернизацию системы курсового и дипломного проектирования, которая должна повышать роль студента в подборе материала, поиске путей решения задач.

Самостоятельная работа приводит студента к получению нового знания, упорядочению и углублению имеющихся знаний, формированию у него профессиональных навыков и умений. Самостоятельная работа выполняет ряд функций:

- развивающую;
- информационно-обучающую;
- ориентирующую и стимулирующую;
- воспитывающую;
- исследовательскую.

В рамках курса выполняются следующие виды самостоятельной работы:

1. Проработка учебного материала (по конспектам, учебной и научной литературе);
2. Выполнение разноуровневых задач и заданий;
3. Работа с тестами и вопросами для самопроверки;
4. Выполнение итоговой контрольной работы.

Студентам рекомендуется с самого начала освоения курса работать с литературой и предлагаемыми заданиями в форме подготовки к очередному аудиторному занятию. При этом актуализируются имеющиеся знания, а также создается база для усвоения нового материала, возникают вопросы, ответы на которые студент получает в аудитории.

Необходимо отметить, что некоторые задания для самостоятельной работы по курсу имеют определенную специфику. При освоении курса студент может пользоваться библиотекой вуза, которая в полной мере обеспечена соответствующей литературой. Значительную помощь в подготовке к очередному занятию может оказать имеющийся в учебно-методическом комплексе краткий конспект лекций. Он же может использоваться и для закрепления полученного в аудитории материала. Самостоятельная работа студентов предусмотрена учебным планом и выполняется в обязательном порядке. Задания предложены по каждой изучаемой теме и могут готовиться индивидуально или в группе. По необходимости студент может обращаться за консультацией к преподавателю. Выполнение заданий контролируется и оценивается преподавателем.

Для успешного самостоятельного изучения материала сегодня используются различные средства обучения, среди которых особое место занимают информационные технологии разного уровня и направленности: электронные учебники и курсы лекций, базы тестовых заданий и задач. Электронный учебник представляет собой программное средство, позволяющее представить для изучения теоретический материал, организовать апробирование, тренаж и самостоятельную творческую работу, помогающее студентам и преподавателю оценить уровень знаний в определенной тематике, а также содержащее необходимую справочную информацию. Электронный учебник может интегрировать в себе возможности различных педагогических программных средств: обучающих программ, справочников, учебных баз данных, тренажеров, контролирующих программ.

Для успешной организации самостоятельной работы все активнее применяются разнообразные образовательные ресурсы в сети Интернет: системы тестирования по различным областям, виртуальные лекции, лаборатории, при этом пользователю достаточно иметь компьютер и подключение к Интернету для того, чтобы связаться с преподавателем, решать вычислительные задачи и получать знания. Использование сетей усиливает роль самостоятельной работы студента и позволяет кардинальным образом изменить методику преподавания.

Студент может получать все задания и методические указания через сервер, что дает ему возможность привести в соответствие личные возможности с необходимыми для выполнения работ трудозатратами. Студент имеет возможность выполнять работу дома или в аудитории. Большое воспитательное и образовательное значение в самостоятельном учебном труде студента имеет самоконтроль. Самоконтроль возбуждает и поддерживает внимание и интерес, повышает

активность памяти и мышления, позволяет студенту своевременно обнаружить и устранить допущенные ошибки и недостатки, объективно определить уровень своих знаний, практических умений. Самое доступное и простое средство самоконтроля с применением информационно-коммуникационных технологий - это ряд тестов «on-line», которые позволяют в режиме реального времени определить свой уровень владения предметным материалом, выявить свои ошибки и получить рекомендации по самосовершенствованию.

Методические рекомендации по работе с литературой

Всю литературу можно разделить на учебники и учебные пособия, оригинальные научные монографические источники, научные публикации в периодической печати. Из них можно выделить литературу основную (рекомендуемую), дополнительную и литературу для углубленного изучения дисциплины.

Изучение дисциплины следует начинать с учебника, поскольку учебник – это книга, в которой изложены основы научных знаний по определенному предмету в соответствии с целями и задачами обучения, установленными программой.

При работе с литературой необходимо учитывать, что имеются различные виды чтения, и каждый из них используется на определенных этапах освоения материала.

Предварительное чтение направлено на выявление в тексте незнакомых терминов и поиск их значения в справочной литературе. В частности, при чтении указанной литературы необходимо подробнейшим образом анализировать понятия.

Сквозное чтение предполагает прочтение материала от начала до конца. Сквозное чтение литературы из приведенного списка дает возможность студенту сформировать свод основных понятий из изучаемой области и свободно владеть ими.

Выборочное – наоборот, имеет целью поиск и отбор материала. В рамках данного курса выборочное чтение, как способ освоения содержания курса, должно использоваться при подготовке к практическим занятиям по соответствующим разделам.

Аналитическое чтение – это критический разбор текста с последующим его конспектированием. Освоение указанных понятий будет наиболее эффективным в том случае, если при чтении текстов студент будет задавать к этим текстам вопросы. Часть из этих вопросов сформулирована в ФОС в перечне вопросов для собеседования. Перечень этих вопросов ограничен, поэтому важно не только содержание вопросов, но сам принцип освоения литературы с помощью вопросов к текстам.

Целью *изучающего* чтения является глубокое и всестороннее понимание учебной информации. Есть несколько приемов изучающего чтения:

1. Чтение по алгоритму предполагает разбиение информации на блоки: название; автор; источник; основная идея текста; фактический материал; анализ текста путем сопоставления имеющихся точек зрения по рассматриваемым вопросам; новизна.

2. Прием постановки вопросов к тексту имеет следующий алгоритм:

- медленно прочитать текст, стараясь понять смысл изложенного;
- выделить ключевые слова в тексте;
- постараться понять основные идеи, подтекст и общий замысел автора.

3. Прием тезирования заключается в формулировании тезисов в виде положений, утверждений, выводов.

К этому можно добавить и иные приемы: прием реферирования, прием комментирования.

Важной составляющей любого солидного научного издания является список литературы, на которую ссылается автор. При возникновении интереса к какой-то обсуждаемой в тексте проблеме всегда есть возможность обратиться к списку относящейся к ней литературы. В этом случае вся проблема как бы разбивается на составляющие части, каждая из которых может изучаться отдельно от других. При этом важно не терять из вида общий контекст и не погружаться чрезмерно в детали, потому что таким образом можно не увидеть главного.

Подготовка к экзамену должна проводиться на основе лекционного материала, материала практических занятий с обязательным обращением к основным учебникам по курсу. Это позволит исключить ошибки в понимании материала, облегчит его осмысление, прокомментирует материал многочисленными примерами.

Методические рекомендации для подготовки к экзамену:

Зачет в VIII-м семестре является формой итогового контроля знаний и умений обучающихся по данной дисциплине, полученных на лекциях, практических занятиях и в процессе самостоятельной работы. Основой для определения оценки служит уровень усвоения обучающимися материала, предусмотренного данной рабочей программой. К зачету допускаются студенты, набравшие 36 и более баллов по итогам текущего и промежуточного контроля. На зачете студент может набрать до 30 баллов.

В период подготовки к зачету обучающиеся вновь обращаются к учебно-методическому материалу и закрепляют промежуточные знания.

Подготовка обучающегося к зачету включает три этапа:

- самостоятельная работа в течение семестра;
- непосредственная подготовка в дни, предшествующие экзамену по темам курса;
- подготовка к ответу на зачетные вопросы.

При подготовке к экзамену обучающимся целесообразно использовать материалы лекций, учебно-методические комплексы, нормативные документы, основную и дополнительную литературу.

На зачет выносится материал в объеме, предусмотренном рабочей программой учебной дисциплины за семестр. Зачет проводится в письменной / устной форме.

При проведении зачета в письменной (устной) форме, ведущий преподаватель составляет экзаменационные билеты, которые включают в себя: тестовые задания; теоретические задания; задачи или ситуации. Формулировка теоретических задания совпадает с формулировкой перечня зачетных вопросов, доведенных до сведения обучающихся накануне экзаменационной сессии. Содержание вопросов одного билета относится к различным разделам программы с тем, чтобы более полно охватить материал учебной дисциплины.

В аудитории, где проводится устный зачет, должно одновременно находиться не более шести студентов на одного преподавателя, принимающего зачет. На подготовку ответа на билет на экзамене отводится 40 минут.

При проведении письменного зачета на работу отводится 60 минут.

Результат устного (письменного) зачета выражается оценками:

Оценка «зачтено» – от 61 до 100 баллов – теоретическое содержание курса освоено, необходимые практические навыки работы сформированы, выполненные учебные задания содержат незначительные ошибки. На экзамене студент демонстрирует твердое знание основного (программного) материала, умеет четко, грамотно, без существенных неточностей отвечать на поставленные вопросы.

Оценка «не зачтено» – от 36 до 60 баллов – теоретическое содержание курса освоено не полностью, необходимые практические навыки работы сформированы частично, выполненные учебные задания содержат грубые ошибки. На зачете студент демонстрирует знание только основного материала, ответы содержат неточности, слабо аргументированы, нарушена последовательность изложения материала

8. Материально-техническое обеспечение дисциплины

8.1. Требования к материально-техническому обеспечению

Для реализации рабочей программы дисциплины имеются специальные помещения для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания оборудования. Специальные помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления информации большой аудитории.

Для проведения занятий лекционного типа имеются демонстрационное оборудование и учебно-наглядные пособия. По дисциплине «Объектно-ориентированное программирование» имеются презентации по отдельным темам курса, позволяющие наиболее эффективно освоить представленный учебный материал.

При проведении занятий лекционного / семинарского типа занятий используются:

лицензионное программное обеспечение:

- Продукты Microsoft (Desktop Education ALNG LicSaPk OLVS Academic Edition Enterprise) подписка (Open Value Subscription);

Антивирусное программное обеспечение Kaspersky Endpoint Security Стандартный Russian Edition;

свободно распространяемые программы:

- Academic MarthCAD License - математическое программное обеспечение, которое позволяет выполнять, анализировать важнейшие инженерные расчеты и обмениваться ими;

- WinZip для Windows - программ для сжатия и распаковки файлов;

- Adobe Reader для Windows – программа для чтения PDF файлов;

- Far Manager - консольный файловый менеджер для операционных систем семейства Microsoft Windows.

- Среда программирования Pascal ABC, Dev C++.

8.2 Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Для студентов с ограниченными возможностями здоровья созданы специальные условия для получения образования. В целях доступности получения высшего образования по образовательным программам инвалидами и лицами с ограниченными возможностями здоровья университетом обеспечивается:

1. Альтернативная версия официального сайта в сети «Интернет» для слабовидящих;

2. Для инвалидов с нарушениями зрения (слабовидящие, слепые)

- присутствие ассистента, оказывающего обучающемуся необходимую помощь, дублирование вслух справочной информации о расписании учебных занятий; наличие средств для усиления остаточного зрения, брайлевской компьютерной техники, видеоувеличителей, программ не-визуального доступа к информации, программ-синтезаторов речи и других технических средств приема-передачи учебной информации в доступных формах для студентов с нарушениями зрения;

- задания для выполнения на экзамене зачитываются ассистентом;

- письменные задания выполняются на бумаге, надиктовываются ассистенту обучающимся;

3. Для инвалидов и лиц с ограниченными возможностями здоровья по слуху (слабослышащие, глухие):

- на зачете/экзамене присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);

- зачет/экзамен проводится в письменной форме;

4. Для инвалидов и лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата, созданы материально-технические условия обеспечивающие возможность беспрепятственного доступа обучающихся в учебные помещения, объекты питания, туалетные и другие помещения университета, а также пребывания в указанных помещениях (наличие расширенных дверных проемов, поручней и других приспособлений).

- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;

- по желанию студента экзамен проводится в устной форме.

Обучающиеся из числа лиц с ограниченными возможностями здоровья обеспечены электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

Специализированные аудитории Института физики и математики КБГУ (219, 220, 254) оснащены персональными компьютерами с доступом к сети интернет.

9. Лист изменений (дополнений)

в рабочую программу по дисциплине «Объектно-ориентированное программирование» по направлению подготовки 01.03.02 – Прикладная математика и информатика; Профиль Математическое и компьютерное моделирование, на 2022-2023 учебный год

№п/п	Элемент (пункт) РПД	Перечень вносимых изменений (дополнений)	Примечание

Обсуждена и рекомендована на заседании кафедры Прикладной математики и информатики протокол № ____ от " ____ " _____ 2022 г.

Заведующий кафедрой _____ / Бечелова А.Р. /

10.Приложения

Приложение 1

Распределение баллов текущего и рубежного контроля

n/n	Вид контроля	Сумма баллов			
		Общая сумма	1-я точка	2-я точка	3-я точка
1-	Посещение занятий	до 10 баллов	до 3 б.	до 3б.	до 4б.
2-	Текущий контроль:	до 30 баллов	до 10 б.	до 10 б.	до 10 б.
	Ответ на 5 вопросов	от 0 до 15 б.	от 0 до 5 б.	от 0 до 5 б.	от 0 до 5 б.
	Полный правильный ответ	до 15 баллов	5 б.	5 б.	5 б.
	Неполный правильный ответ	от 3 до 15 б.	от 1 до 5 б.	от 1 до 5 б.	от 1 до 5 б.
	Ответ, содержащий неточности, ошибки	0б.	0б.	0б.	0б.
	Выполнение самостоятельных заданий (решение задач, написание рефератов, доклад, эссе)	от 0 до 15 б.	от 0 до 5 б.	от 0 до 5 б.	от 0 до 5 б.
1.	Рубежный контроль	до 30 баллов	до 10 б.	до 10 б.	до 10 б.
	Тестирование	от 0- до 12б.	от 0- до 4б.	от 0- до 4б.	от 0- до 4б.
	Коллоквиум	от 0 до 18б.	от 0 до 6 б.	от 0 до 6 б.	от 0 до 6 б.
	Итого сумма текущего и рубежного контроля	до 70баллов	до 23б.	до 23б	до 24б
	Первый этап (базовый уровень) – оценка «удовлетворительно»	не менее 36 б.	не менее 12 б.	не менее 12 б	не менее 12 б
	Второй этап (продвинутый уровень) – оценка «хорошо»	менее 70 б. (51-69 б.)	менее 23 б	менее 23 б	менее 24б
	Третий этап (высокий уровень) - оценка «отлично»	не менее 70 б.	не менее 23 б.	не менее 23 б	не менее 24б

Шкала оценивания планируемых результатов обучения**Текущий и рубежный контроль**

Семестр	Шкала оценивания			
	0-35 баллов	36-50 баллов	51-60 баллов	56-70 баллов
8	Частичное посещение аудиторных занятий. Неудовлетворительное выполнение лабораторных и практических работ. Плохая подготовка к балльно-рейтинговым мероприятиям. Студент не допускается к промежуточной аттестации	Полное или частичное посещение аудиторных занятий. Частичное выполнение и защита лабораторных и практических работ. Выполнение контрольных работ, тестовых заданий, ответы на коллоквиуме на оценки «удовлетворительно».	Полное или частичное посещение аудиторных занятий. Полное выполнение и защита лабораторных и практических работ. Выполнение контрольных работ, тестовых заданий, ответы на коллоквиуме на оценки «хорошо».	Полное посещение аудиторных занятий. Полное выполнение и защита лабораторных и практических занятий. Выполнение контрольных работ, тестовых заданий, ответы на коллоквиуме на оценки «отлично».

Промежуточная аттестация (для зачёта)

Семестр	Шкала оценивания	
	Незачтено (36-60)	Зачтено (61-70)
8	Студент имеет 36-60 баллов по итогам текущего и рубежного контроля, на зачёте не ответил ни на один вопрос.	Студент имеет 36-45 баллов по итогам текущего и рубежного контроля, на зачете представил полный ответ на один вопрос и частично (полностью) ответил на второй. Студент имеет 46-60 баллов по итогам текущего и рубежного контроля, на зачете дал полный ответ на один вопрос или частично ответил на оба вопроса. Студенту, имеющему 61-70 баллов по итогам текущего и рубежного контроля, выставляется отметка «зачтено» без сдачи зачёта.