

ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»
ИМЕНИ В.Ф. УТКИНА

Кафедра «Вычислительная и прикладная математика»

**МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ
«Объектно-ориентированное программирование»**

Направление подготовки

09.05.01 «Применение и эксплуатация автоматизированных систем
специального назначения»

Направленность (профиль) подготовки

Математическое, программное и информационное обеспечение вычислительной
техники и автоматизированных систем

Квалификация выпускника - специалист

Форма обучения - очная

Рязань

1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ.

Лабораторная работа №1

Структуры и модули

Задание: В данной лабораторной работе необходимо создать модуль, содержащий описание структуры данных, представляющее собой некоторую сущность: вектор, дробь, фигура и т.д. (по вариантам выданным преподавателем), а также 8 операций (функций) над этой структурой. Основная программа должна запрашивать у пользователя какую операцию необходимо протестировать, далее запрашивать данные для операндов и показывать результат операции с помощью функции модуля «вывод в консоль». В качестве упрощения, задание не подразумевает использование динамической памяти, следовательно, во всех вариантах используются статические массивы (если это обусловлено задачей). Важно: структуры создаются в модуле с помощью отдельной функции, возвращающей структуру по значению. Это необходимо, например, для того чтобы рассчитать некоторые поля структуры на основе переданных данных.

Рекомендуемая литература: Объектно-ориентированное программирование: метод. указ. к лаб. работам и практ. занятиям / Рязан. гос. радиотехн. универ.; Сост.: О.В. Антипов – Рязань, 2021.

Лабораторная работа №2

Простые классы

Задание: В данной лабораторной работе необходимо преобразовать лабораторную работу №1 таким образом, чтобы заданная сущность была представлена в виде класса. Класс должен быть обязательно оформлен отдельными файлами описания (.h) и реализации (.cpp). Данный класс должен иметь конструктор с параметрами, необходимыми для создания сущности, методы для реализации всех нужных операций над сущностью. Доступ к полям класса должен быть приватным, извне доступ к ним осуществляется исключительно через геттеры и сеттеры класса. Также в данном задании необходимо написать класс юнит-тестов для каждого нетривиального метода созданного класса. Количество тестов должно быть не менее 10. Результаты работы юнит-тестов необходимо выводить в консоль. Функция main должна содержать только запуск всех юнит-тестов.

Рекомендуемая литература: Объектно-ориентированное программирование: метод. указ. к лаб. работам и практ. занятиям / Рязан. гос. радиотехн. универ.; Сост.: О.В. Антипов – Рязань, 2021.

Лабораторная работа №3

Наследование, статические методы, полиморфизм

Задание: В данной лабораторной работе необходимо изменить класс, созданный в лабораторной работе №2 следующим образом: добавить для методов параметры-ссылки там, где это необходимо, добавить константные методы и параметры там, где это оправданно, заменить методы, реализующие операции над сущностью, перегруженными операторами (соответствующими смыслу операции), добавить статический метод для класса, возвращающий новый объект, заполненный случайными значениями из заданного диапазона (который задаётся через параметры данного метода). Для бинарных значений диапазон не указывается, добавить наследование от базового класса Object, в котором должна быть реализация следующего функционала: подсчёт количества созданных объектов и количества активных объектов на данный момент через статические поля класса. Статический метод printTotalInfo для вывода этой информации в консоль, функционал для создания динамического массива (через malloc или через new) – списка всех выполненных операций с

объектом. При выполнении каждой операции её обозначение заносится в данный массив через метод `addOp`, метод `clearOp` – очищает массив операций, метод `printOp` – выводит полный список всех выполненных операций с объектом, конструктор копии и оператор присваивания (`=`) для класса `Object`, дополнить класс юнит-тестирования минимум 5 тестами для проверки работы базового класса `Object` и статического метода для случайной генерации класса сущности.

Рекомендуемая литература: Объектно-ориентированное программирование: метод. указ. к лаб. работам и практ. занятиям / Рязан. гос. радиотехн. универ.; Сост.: О.В. Антипов – Рязань, 2021.

Лабораторная работа №4

Полиморфизм. Абстрактные классы.

Задание: В данной лабораторной работе необходимо разработать классы в соответствии со своей предметной областью (варианты выдаются преподавателем). Все задания состоят из набора сущностей, которые будет необходимо оформить в правильные отношения между объектами. В вариантах заданий описаны только обязательные сущности, можно добавлять свои по необходимости, в каждом задании есть сущность базового абстрактного класса, которую реализуют классы-наследники. Также есть композитная сущность – это класс, который владеет односвязным списком указателей на базовый абстрактный класс, может добавлять и удалять экземпляры, а также выполнять другие операции над ними (через полиморфный вызов методов). При добавлении/удалении объектов используется динамическое выделение памяти через `newdelete`. Список необходимо реализовать вручную, через указатели. В программе необходимо предусмотреть возможность случайной генерации `N` объектов-сущностей из варианта задания и их добавления в композитную сущность, возможность удаления объектов по указанному номеру из списка, а также отображения результатов выполнения действий над объектами.

Рекомендуемая литература: Объектно-ориентированное программирование: метод. указ. к лаб. работам и практ. занятиям / Рязан. гос. радиотехн. универ.; Сост.: О.В. Антипов – Рязань, 2021.

Лабораторная работа №5

Шаблонные классы

Задание: В данной лабораторной работе необходимо разработать шаблонный класс в соответствии с вариантом задания (варианты выдаются преподавателем). Любой класс должен иметь конструктор копии и перегруженный оператор `=`. Также класс должен содержать перегрузку оператора `<<` для класса `iostream`, чтобы его можно было выводить в консоль через стандартный поток вывода `cout` с указанием количества элементов и их значений. В главной функции `main` обеспечить консольный интерфейс для тестирования всех функций шаблонного класса с типами `int`, `float`, `char*`, `struct Vec2 float x; float y;`

Рекомендуемая литература: Объектно-ориентированное программирование: метод. указ. к лаб. работам и практ. занятиям / Рязан. гос. радиотехн. универ.; Сост.: О.В. Антипов – Рязань, 2021.

Лабораторная работа №6

Стандартные контейнерные классы, исключения

Задание: В данной лабораторной работе необходимо модифицировать программу из лабораторной работы №4, добавив в неё исключения во всех необходимых местах (не менее 8 случаев использования), также заменить все места использования строковых/числовых констант для идентификации типов объектов на `enum class`, и заменить все рукописные

контейнерные классы, на класс `std::vector`. Для манипуляции с элементами `vector`-а использовать исключительно итератор

Рекомендуемая литература: Объектно-ориентированное программирование: метод. указ. к лаб. работам и прак. занятиям / Рязан. гос. радиотехн. универ.; Сост.: О.В. Антипов – Рязань, 2021.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ.

Практическое занятие №1

Вопросы:

1. Что такое объектно-ориентированное программирование? Какова цель внедрения ООП?
2. Что такое объект и что такое класс? Дайте определение принципа ООП абстракции и инкапсуляции.
3. Что такое объект и что такое класс? Дайте определение принципа ООП наследования и полиморфизма.
4. Какие достоинства и недостатки ООП вы знаете?
5. Дайте краткие сведения о языке C++. Чем он отличается от C?
6. Для чего нужны пространства имен и как их объявлять? Для чего нужны `using`-объявления и как их использовать?
7. Что такое потоки ввода-вывода и как с ними работать в C++?
8. Какие уровни доступа есть при объявлении класса? Формат объявления класса.

Задание:

Создайте программу на языке C++, которая моделирует систему управления учебным заведением. Разработайте классы "Студент" и "Преподаватель". Класс "Студент" должен содержать следующие атрибуты: имя, возраст, средний балл. Реализуйте методы для получения и установки значений каждого атрибута. Определите уровень доступа к атрибутам класса "Студент" таким образом, чтобы они были доступны только внутри класса. Класс "Преподаватель" должен содержать следующие атрибуты: имя, возраст, предмет. Реализуйте методы для получения и установки значений каждого атрибута. Определите уровень доступа к атрибутам класса "Преподаватель" таким образом, чтобы они были доступны только внутри класса и его производных классов. Создайте объекты классов "Студент" и "Преподаватель" в основной программе. Попробуйте получить доступ к атрибутам классов из основной программы и обратите внимание на ошибки доступа, которые возникают из-за уровней доступа. Объясните в комментариях к программе, какие уровни доступа используются при объявлении классов и почему они выбраны.

Практическое занятие №2

Вопросы:

1. Что такое конструктор и деструктор? Формат их объявления.
2. Что такое указатель `this` и в каких случаях он применяется?
3. Зачем нужна интерфейсная часть и часть реализации при объявлении класса в программе?
4. Какие рекомендации существуют по объявлению классов?
5. Что такое юнит-тестирование? На какие части обычно разделен код теста?
6. Что такое перегрузка функций? Как компилятор определяет какая функция вызывается? Зачем был введен `nullptr` и в чём его отличие от `NULL`?
7. Что такое конструктор копирования? В каких случаях он объявляется явно?
8. Что такое ссылки и в чём их отличие от указателей? В каких случаях надо применять ссылки, а в каких указатели?

Задание:

Создайте класс "Студент", который будет представлять собой объектно-ориентированную модель студента. Включите в класс следующие атрибуты: имя, возраст, средний балл. Реализуйте конструктор, деструктор и методы для получения и установки значений каждого атрибута. В конструкторе класса "Студент" инициализируйте атрибуты объекта с помощью параметров конструктора. В деструкторе класса "Студент" выведите сообщение о том, что объект класса "Студент" уничтожен. В классе "Студент" объявите метод, который будет выводить информацию о студенте, используя указатель `this` для доступа к атрибутам объекта. Создайте интерфейсную часть класса "Студент", объявив публичные методы для доступа к атрибутам объекта. Создайте часть реализации класса "Студент", где определите методы для получения и установки значений атрибутов. В основной программе создайте объект класса "Студент" и продемонстрируйте использование методов для работы с атрибутами объекта. Напишите юнит-тесты для класса "Студент", проверяющие корректность работы методов для получения и установки значений атрибутов. Разделите код теста на три части: инициализация объекта, вызов методов и проверка результатов. Перегрузите функцию вывода на экран для класса "Студент", чтобы она выводила информацию о студенте. Объясните, как компилятор определяет, какая функция вызывается в зависимости от типа аргументов. Объявите явно конструктор копирования для класса "Студент", который будет создавать новый объект "Студент" на основе существующего объекта. Объясните, в каких случаях необходимо объявлять конструктор копирования явно. Объявите методы класса "Студент" с использованием ссылок и указателей. Объясните, в каких случаях следует использовать ссылки, а в каких указатели.

Практическое занятие №3

Вопросы:

1. Зачем используются ссылки как параметры функции? А в качестве возвращаемого значения?
2. Что такое статические поля и методы класса? Зачем они используются?
3. Что такое перегрузка операторов и какие её виды бывают? Как она объявляется? Какие операторы нельзя перегружать?
4. Что такое константные методы класса и для чего они нужны?
5. Что такое наследование и для чего оно нужно? Что такое суперкласс? Подкласс? Базовый класс?
6. В каком порядке при наследовании вызываются конструкторы и деструкторы? Какие типы наследования бывают?
7. Что такое переопределение метода и зачем оно используется? Зачем нужно ключевое слово `auto`?
8. С помощью каких операторов происходит выделение/освобождение динамической памяти в C++? В чём их преимущество перед функциями из C?
9. Что такое дружественная функция и класс? Когда их стоит использовать?

Задание:

Создайте программу на языке C++, которая моделирует систему управления библиотекой. Разработайте классы "Книга" и "Библиотека". Класс "Книга" должен содержать информацию о названии, авторе и годе издания книги. Класс "Библиотека" должен иметь функционал для добавления книги в коллекцию, удаления книги из коллекции и вывода списка всех книг в библиотеке. Используйте ссылки в качестве параметров функций для передачи объектов классов "Книга" и "Библиотека". Также используйте ссылки в качестве возвращаемого значения функции для возврата объектов класса "Книга" из методов класса "Библиотека". Создайте класс "Студент" с статическим полем "количествоСтудентов" и статическим методом

"получитьКоличествоСтудентов". Статическое поле "количествоСтудентов" должно хранить общее количество созданных объектов класса "Студент". Статический метод "получитьКоличествоСтудентов" должен возвращать значение статического поля "количествоСтудентов". Объясните, зачем используются статические поля и методы класса. Создайте класс "Вектор" с перегруженными операторами сложения, вычитания и умножения. Перегрузите операторы для работы с объектами класса "Вектор". Объясните, что такое перегрузка операторов и какие бывают её виды. Объясните, как объявляется перегрузка операторов и какие операторы нельзя перегружать. Создайте класс "Круг" с константным методом "вычислитьПлощадь". Константный метод "вычислитьПлощадь" должен возвращать площадь круга и не изменять состояние объекта класса "Круг". Объясните, что такое константные методы класса и для чего они нужны. Создайте класс "Фигура" и класс "Прямоугольник", который наследуется от класса "Фигура". Объясните, что такое наследование и для чего оно нужно. Объясните понятия суперкласса, подкласса и базового класса.

Практическое занятие №4

Вопросы:

1. Что такое полиморфизм? Какие виды полиморфизма бывают?
2. Что такое полиморфизм? Что такое полиморфизм подтипов? Какой класс называют полиморфным?
3. Что такое виртуальная функция? Как добавить определение по умолчанию для виртуальной функции?
4. Что такое виртуальная функция? В чём отличие виртуальных функций от обычного переопределения?
5. Для чего используется виртуальный деструктор? Для чего нужны модификаторы `override` и `final`?
6. Что такое чистые виртуальные функции? Что такое абстрактный класс и для чего он используется?
7. Что такое интерфейсный класс и для чего он используется?
8. Что такое тип связи композиция? Приведите пример.

Задание:

Создайте класс "Фигура", который представляет абстрактную геометрическую фигуру. Включите в класс следующие атрибуты: цвет и площадь. Реализуйте методы для получения и установки значений каждого атрибута. Создайте производные классы "Круг" и "Прямоугольник", которые наследуются от класса "Фигура". Добавьте в каждый из этих классов методы для вычисления площади соответствующей фигуры. Создайте класс "ФигурыМенеджер", который будет управлять коллекцией фигур. Включите в класс следующие методы: Метод для добавления фигуры в коллекцию. Метод для вычисления общей площади всех фигур в коллекции. Метод для вывода информации о каждой фигуре в коллекции. Создайте объекты классов "Круг" и "Прямоугольник" и добавьте их в объект класса "ФигурыМенеджер". Вычислите общую площадь всех фигур и выведите информацию о каждой фигуре. Реализуйте дружественную функцию "сравнитьПлощади", которая принимает два объекта класса "Фигура" и сравнивает их площади. Функция должна выводить сообщение о том, какая фигура имеет большую площадь. Создайте объекты классов "Круг" и "Прямоугольник" и используйте дружественную функцию "сравнитьПлощади" для сравнения их площадей. Создайте абстрактный класс "ИнтерфейсФигуры", который будет представлять интерфейс для работы с фигурами. Включите в класс чисто виртуальные методы для получения площади и вывода информации о фигуре. Измените класс "Фигура" так, чтобы он стал наследником абстрактного класса "ИнтерфейсФигуры". Реализуйте в классе "Фигура"

чисто виртуальные методы из интерфейса. Создайте объекты классов "Круг" и "Прямоугольник" и используйте указатель на абстрактный класс "ИнтерфейсФигуры" для работы с ними. Вычислите площадь каждой фигуры и выведите информацию о них.

Практическое занятие №5

Вопросы:

1. Что такое тип связи агрегация? Приведите пример.
2. Что такое тип связи ассоциация? Приведите пример. 36. Что такое тип связи зависимость? Приведите пример.
3. Что такое тип связи наследование? Приведите пример.
4. Что такое контейнерные классы и какие виды их бывают?
5. Что такое множественное наследование? Что такое ромбовидное наследование и какие проблемы с этим связаны?
6. Что такое множественное наследование? Что такое виртуальное наследование и для чего оно применяется в языке C++?
7. Что такое шаблоны и для чего они применяются? Как объявить шаблон функции и шаблон класса?
8. Что такое инстанцирование шаблона и в какой момент оно происходит? Какие преимущества и недостатки есть у шаблонов?

Задание:

Создайте класс "Человек" с атрибутами "имя" и "возраст". Реализуйте методы для получения и установки значений каждого атрибута. Создайте класс "Дом", который представляет собой контейнер для объектов класса "Человек". Реализуйте методы для добавления и удаления людей из дома, а также для получения списка всех людей в доме. Создайте класс "Улица", который представляет собой контейнер для объектов класса "Дом". Реализуйте методы для добавления и удаления домов на улице, а также для получения списка всех домов на улице. Создайте шаблон класса "Контейнер", который представляет собой обобщенный контейнер для хранения объектов любого типа. Реализуйте методы для добавления и удаления объектов из контейнера, а также для получения списка всех объектов в контейнере. В программе создайте объекты классов "Человек", "Дом" и "Улица". Добавьте людей в дом, добавьте дом на улицу, а затем добавьте улицу в контейнер. Продемонстрируйте использование методов для получения информации о людях, домах и улицах. Расширьте класс "Человек" и класс "Дом" добавлением дополнительных атрибутов и методов по вашему выбору. Расширьте шаблон класса "Контейнер" добавлением дополнительных методов по вашему выбору. Объясните, какие типы связи присутствуют между классами "Человек", "Дом", "Улица" и "Контейнер".

Практическое занятие №6

Вопросы:

1. Что такое non-тире параметры шаблона, какого типа они могут быть? Что такое явная специализация шаблона и каким образом она объявляется?
2. Что такое частичная специализация шаблона и каким образом она объявляется? Какое ограничение существует у частичной специализации шаблона и каким образом его можно обходить?
3. Чем характеризуется приведение типов с помощью `static_cast` и `dynamic_cast`?
4. Чем характеризуется приведение типов с помощью `const_cast` и `reinterpret_cast`?
5. Что такое `enum class` и в чём его отличие от обычного `enum`?
6. Что такое исключения? В чём преимущество исключений перед возвратом кодов ошибок?
7. Что такое исключения? Какова общая структура выбрасывания и обработки исключения? Из каких блоков она состоит?

8. Что такое исключения? Для чего нужны классы-исключений? Каковы недостатки исключений?

9. Что такое исключения? В каких случаях стоит использовать исключения?

Задание:

Создайте шаблон класса "Стек", который будет представлять собой структуру данных стек. Реализуйте методы для добавления элемента в стек, удаления элемента из стека и получения верхнего элемента стека. Шаблон класса должен иметь параметр типа данных, который будет определять тип элементов, хранящихся в стеке. В основной программе создайте объект класса "Стек" с типом данных "int" и добавьте в него несколько элементов. Затем попробуйте удалить элемент из пустого стека и получить верхний элемент из пустого стека. Обработайте исключения, которые могут возникнуть при выполнении этих операций. Добавьте в класс "Стек" метод, который будет возвращать размер стека. Попробуйте получить размер стека, когда стек пустой, и обработайте исключение, которое может возникнуть при этой операции. Создайте класс "Фигура", который будет представлять собой абстрактную фигуру. Реализуйте в этом классе виртуальный метод "площадь", который будет возвращать площадь фигуры. Создайте производные классы "Круг", "Прямоугольник" и "Треугольник", которые будут наследоваться от класса "Фигура" и переопределять метод "площадь" для вычисления площади соответствующей фигуры. В основной программе создайте объекты классов "Круг", "Прямоугольник" и "Треугольник" и выведите на экран их площади. Обработайте исключение, которое может возникнуть при вызове метода "площадь" для объекта класса "Фигура".

Практическое занятие №7

Вопросы:

1. Что такое STL и из каких частей она состоит? Что такое последовательные контейнеры?
2. Что такое STL и из каких частей она состоит? Какие есть стандартные последовательные контейнеры в STL?
3. Что такое итераторы и для чего они применяются?
4. Какие перегруженные операторы имеют итераторы?
5. Какие есть основные методы для работы с итераторами?
6. Какие типы итераторов обычно предоставляют контейнеры?
7. Что такое STL и из каких частей она состоит? Что такое ассоциативные контейнеры?
8. Что такое STL и из каких частей она состоит? Какие есть стандартные ассоциативные контейнеры в STL?

Задание:

Создайте класс "Студент" со следующими атрибутами: имя (строка), возраст (целое число) и средний балл (вещественное число). Создайте контейнер типа "vector" для хранения объектов класса "Студент". Реализуйте функцию "добавитьСтудента", которая принимает имя, возраст и средний балл студента в качестве параметров и добавляет новый объект класса "Студент" в контейнер. Реализуйте функцию "найтиСтудента", которая принимает имя студента в качестве параметра и возвращает указатель на объект класса "Студент" с соответствующим именем. Если студент не найден, функция должна возвращать nullptr. Реализуйте функцию "удалитьСтудента", которая принимает имя студента в качестве параметра и удаляет объект класса "Студент" с соответствующим именем из контейнера. Реализуйте функцию "вывестиСтудентов", которая выводит на экран информацию о всех студентах в контейнере. В основной программе создайте меню, в котором пользователь может выбрать одну из следующих операций: Добавить студента Найти студента Удалить студента Вывести список студентов Выход При выборе операции "Добавить студента", программа должна запрашивать у пользователя имя, возраст и средний балл студента, а затем вызывать функцию "добавитьСтудента". При выборе операции "Найти студента", программа должна запрашивать

у пользователя имя студента, а затем вызывать функцию "найтиСтудента" и выводить информацию о найденном студенте на экран. При выборе операции "Удалить студента", программа должна запрашивать у пользователя имя студента, а затем вызывать функцию "удалитьСтудента". При выборе операции "Вывести список студентов", программа должна вызывать функцию "вывестиСтудентов". При выборе операции "Выход", программа должна завершаться.

Практическое занятие №8

Вопросы:

1. Что такое адаптеры? Какие есть стандартные адаптеры в STL?
2. Что такое алгоритмы STL и как они реализованы?
3. Что такое объект `std::initializer_list` и для чего он используется?
4. Что такое объект `std::string` и какие основные методы имеет?
5. Что такое идиома RAII?
6. Что такое умный указатель и для чего он используется?
7. Какие рекомендуемые умные указатели есть в стандартной библиотеке?
8. Что такое циклическая зависимость, к каким проблемам она может приводить и какие пути решения существуют?
9. Что такое лямбда-выражения и какой формат их объявления?

Задание:

Создайте класс "Task" с приватными атрибутами "название" (`std::string`) и "статус" (`bool`). Реализуйте конструктор, деструктор и методы для получения и установки значений атрибутов. Создайте класс "TaskManager", который будет управлять списком задач. В классе "TaskManager" объявите приватный контейнер `std::vector` для хранения объектов класса "Task". Реализуйте методы класса "TaskManager" для добавления новой задачи, удаления задачи по названию, изменения статуса задачи и вывода списка всех задач. Используйте ассоциативный контейнер `std::map` для хранения объектов класса "TaskManager". Ключом в контейнере будет название задачи, а значением - объект класса "TaskManager". Это позволит вам иметь несколько списков задач с разными названиями. Реализуйте методы класса "TaskManager" для добавления нового списка задач, удаления списка задач по названию и вывода списка всех списков задач. Используйте умные указатели `std::unique_ptr` или `std::shared_ptr` для управления памятью объектов классов "Task" и "TaskManager". Это позволит автоматически освобождать память при удалении объектов или при выходе из области видимости. Напишите основную программу, в которой пользователь может выбрать операции для управления списком задач и списками задач.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К КУРСОВОЙ РАБОТЕ.

Задание №1. Создать класс для управления процессом управления рисками. Класс должен иметь методы для управления рисками, включая идентификацию рисков, оценку рисков, управление рисками и т.д. Можно использовать концепции наследования и полиморфизма для создания различных типов рисков (например, финансовые риски, технические риски и т.д.).

Задание №2. Создать класс для управления проектами. Класс должен иметь методы для управления проектами, включая создание проекта, назначение задач, управление сроками и т.д. Можно использовать концепции наследования и полиморфизма для создания различных типов проектов (например, IT-проекты, строительные проекты и т.д.).

Задание №3. Создать класс для управления процессом управления ресурсами. Класс должен

иметь методы для управления ресурсами, включая управление временем, управление материальными ресурсами, управление информационными ресурсами и т.д. Можно использовать концепции наследования и полиморфизма для создания различных типов ресурсов (например, временные ресурсы, материальные ресурсы и т.д.).

Задание №4. Создать класс для управления автомобилем. Класс должен иметь методы для управления скоростью, направлением движения, торможением и т.д. Можно использовать концепции наследования и полиморфизма для создания различных типов автомобилей (например, легковые автомобили, грузовики и т.д.).

Задание №5. Создать класс для управления библиотекой. Класс должен иметь методы для добавления новых книг, удаления книг, изменения информации о книгах и поиска книг по различным критериям (например, по автору, названию, жанру и т.д.). Можно использовать концепции наследования и полиморфизма для создания различных типов книг (например, художественная литература научно-популярная литература и т.д.).

Задание №6. Создать класс для управления процессом обучения. Класс должен иметь методы для добавления студентов, удаления студентов, изменения информации о студентах и поиска студентов по различным критериям (например, по курсу, группе, успеваемости и т.д.). Можно использовать концепции наследования и полиморфизма для создания различных типов курсов (например, онлайн-курсы, очные курсы и т.д.).

Задание №7. Создать класс для управления базой данных студентов. Класс должен иметь методы для добавления новых студентов, удаления студентов, изменения информации о студентах и поиска студентов по различным критериям (например, по имени, фамилии, году поступления и т.д.). Можно использовать концепции наследования и полиморфизма для создания различных типов студентов (например, бакалавры, магистры и т.д.).

Задание №8. Создать класс для управления заказами. Класс должен иметь методы для добавления новых заказов, удаления заказов, изменения информации о заказах и поиска заказов по различным критериям (например, по дате, статусу, клиенту и т.д.). Можно использовать концепции наследования и полиморфизма для создания различных типов заказов (например, заказы на продукты, заказы на услуги и т.д.).

Задание №9. Создать класс для управления процессом доставки. Класс должен иметь методы для управления процессом доставки, включая планирование доставки, управление маршрутами, контроль качества и т.д. Можно использовать концепции наследования и полиморфизма для создания различных типов доставки (например, доставка продуктов питания, доставка электроники и т.д.).

Задание №10. Создать класс для управления бизнес-процессами. Класс должен иметь методы для управления процессами, включая начало и окончание процесса, управление задачами, обработку ошибок и т.д. Можно использовать концепции наследования и полиморфизма для создания различных типов бизнес-процессов (например, процессы производства, процессы продаж и т.д.).

Задание №11. Создать класс для управления банковским счетом. Класс должен иметь методы для открытия нового счета, закрытия счета, внесения и снятия денег со счета, а также для проверки баланса. Можно использовать концепции наследования и полиморфизма для

создания различных типов счетов (например, сберегательный счет, текущий счет и т.д.).

Задание №12. Создать класс для управления процессом управления персоналом. Класс должен иметь методы для управления персоналом, включая управление наймом, управление обучением, управление оценкой производительности и т.д. Можно использовать концепции наследования и полиморфизма для создания различных типов персонала (например, менеджеры, специалисты по продажам и т.д.).

Задание №13. Создать класс для управления процессом продаж. Класс должен иметь методы для управления процессом продаж, включая управление заказами, управление складом, контроль качества и т.д. Можно использовать концепции наследования и полиморфизма для создания различных типов продукции (например, продукты питания, электроника и т.д.).

Задание №14. Создать класс для управления складом. Класс должен иметь методы для добавления новых товаров, удаления товаров, изменения информации о товарах и поиска товаров по различным критериям (например по наименованию, производителю, цене и т.д.). Можно использовать концепции наследования и полиморфизма для создания различных типов товаров (например, продукты питания, электроника и т.д.).

Задание №15. Создать класс для управления процессом управления проектами. Класс должен иметь методы для управления процессом управления проектами, включая планирование проекта, управление ресурсами, контроль качества и т.д. Можно использовать концепции наследования и полиморфизма для создания различных типов проектов (например, IT-проекты, строительные проекты и т.д.).

Задание №16. Создать класс для управления процессом обслуживания клиентов. Класс должен иметь методы для управления процессом обслуживания клиентов, включая управление заявками, управление обращениями, контроль качества и т.д. Можно использовать концепции наследования и полиморфизма для создания различных типов обслуживания (например, обслуживание клиентов в сфере телекоммуникаций, обслуживание клиентов в сфере банковского дела и т.д.).

Задание №17. Создать класс для управления процессом управления финансами. Класс должен иметь методы для управления финансами, включая управление бюджетом, управление налогами, управление инвестициями и т.д. Можно использовать концепции наследования и полиморфизма для создания различных типов финансовых операций (например, операции с акциями, операции с облигациями и т.д.).

4. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Перед началом изучения дисциплины студенту необходимо ознакомиться с содержанием рабочей программы дисциплины, с целями и задачами дисциплины, ее связями с другими дисциплинами образовательной программы, методическими разработками по данной дисциплине.

Методические рекомендации студентам по работе над конспектом лекций

Основу теоретического обучения студентов составляют лекции. Они дают систематизированные знания студентам о наиболее сложных и актуальных проблемах

изучаемой дисциплины. На лекциях особое внимание уделяется не только усвоению студентами изучаемых проблем, но и стимулированию их активной познавательной деятельности, творческого мышления, развитию научного мировоззрения, профессионально-значимых свойств и качеств.

Перед каждой лекцией студенту необходимо просматривать рабочую программу дисциплины, что позволит сэкономить время на записывание темы лекции, ее основных вопросов, рекомендуемой литературы.

Перед очередной лекцией необходимо просмотреть по конспекту материал предыдущей лекции. При затруднениях в восприятии материала следует обратиться к основным литературным источникам. Если разобраться в материале опять не удалось, то обратитесь к лектору (по графику его консультаций) или к преподавателю на практических занятиях или лабораторных работах. Не оставляйте «белых пятен» в освоении материала.

Во время лекции студенты должны не только внимательно воспринимать действия преподавателя, но и самостоятельно мыслить, добиваться понимания изучаемого предмета. Студенты должны аккуратно вести конспект. В случае недопонимания какой-либо части предмета следует задать вопрос в установленном порядке преподавателю. В процессе работы на лекции необходимо так же выполнять в конспектах модели изучаемого предмета (рисунки, схемы, чертежи и т. д.), которые использует преподаватель.

Слушая лекцию, нужно из всего получаемого материала выбирать и записывать самое главное. Следует знать, что главные положения лекции преподаватель обычно выделяет интонацией или повторяет несколько раз. Именно поэтому предварительная подготовка к лекции позволит студенту уловить тот момент, когда следует перейти к конспектированию, а когда можно просто внимательно слушать лекцию. В связи с этим нелишне перед началом промежуточной аттестации еще раз бегло просмотреть учебники или прежние конспекты по изучаемым предметам. Это станет первичным знакомством с тем материалом, который прозвучит на лекции, а также создаст необходимый психологический настрой.

Чтобы правильно и быстро конспектировать лекцию важно учитывать, что способы подачи лекционного материала могут быть разными. Преподаватель может диктовать материал, рассказывать его, не давая ничего под запись, либо проводить занятие в форме диалога со студентами. Чаще всего можно наблюдать соединение двух или трех вышеназванных способов.

Эффективность конспектирования зависит от умения владеть правильной методикой записи лекции. Конечно, способы конспектирования у каждого человека индивидуальны. Однако существуют некоторые наиболее употребляемые и целесообразные приемы записи лекционного материала.

Запись лекции можно вести в виде тезисов – коротких, простых предложений, фиксирующих только основное содержание материала. Количество и краткость тезисов может определяться как преподавателем, так и студентом. Естественно, что такая запись лекции требует впоследствии обращения к дополнительной литературе. На отдельные лекции можно приносить соответствующий иллюстративный материал на бумажных или электронных носителях, представленный лектором на портале или присланный на «электронный почтовый ящик группы» (таблицы, графики, схемы). Данный материал будет охарактеризован, прокомментирован, дополнен непосредственно на лекции.

Кроме тезисов важно записывать примеры, доказательства, даты и цифры, имена. Значительно облегчают понимание лекции те схемы и графики, которые вычерчивает на доске преподаватель. По мере возможности студенты должны переносить их в тетрадь рядом с тем текстом, к которому эти схемы и графики относятся.

Хорошо если конспект лекции дополняется собственными мыслями, суждениями, вопросами, возникающими в ходе прослушивания содержания лекции. Те вопросы, которые возникают у

студента при конспектировании лекции, не всегда целесообразно задавать сразу при их возникновении, чтобы не нарушить ход рассуждений преподавателя. Студент может попытаться ответить на них сам в процессе подготовки к семинарам либо обсудить их с преподавателем на консультации.

Важно и то, как будет расположен материал в лекции. Если запись тезисов ведется по всей строке, то целесообразно отделять их время от времени красной строкой или пропуском строки. Примеры же и дополнительные сведения можно смещать вправо или влево под тезисом, а также на поля. В тетради нужно выделять темы лекций, записывать рекомендуемую для самостоятельной подготовки литературу, внести фамилию, имя и отчество преподавателя. Наличие полей в тетради позволяет не только получить «ровный» текст, но и дает возможность при необходимости вставить важные дополнения и изменения в конспект лекции.

При составлении конспектов необходимо использовать основные навыки стенографии. Так в процессе совершенствования навыков конспектирования лекций важно выработать индивидуальную систему записи материала, научиться рационально сокращать слова и отдельные словосочетания.

Практика показывает, что не всегда студенту удается успевать записывать слова лектора даже при использовании приемов сокращения слов. В этом случае допустимо обратиться к лектору с просьбой повторить сказанное. При обращении важно четко сформулировать просьбу, указать какой отрывок необходимо воспроизвести еще раз. Однако не всегда удобно прерывать ход лекции. В этом случае можно оставить пропуск, и после лекции устранить его при помощи конспекта соседа. Важно сделать это в короткий срок, пока свежа память о воспринятой на лекции информации.

Работу над конспектом следует начинать с его доработки, желательно в тот же день, пока материал еще легко воспроизводим в памяти (через 10 часов после лекции в памяти остается не более 30-40 % материала). С целью доработки необходимо прочитать записи, восстановить текст в памяти, а также исправить описки, расшифровать не принятые ранее сокращения, заполнить пропущенные места, понять текст, вникнуть в его смысл. Далее следует прочитать материал по рекомендуемой литературе, разрешая в ходе чтения возникшие ранее затруднения, вопросы, а также дополняя и исправляя свои записи. Записи должны быть наглядными, для чего следует применять различные способы выделений. В ходе доработки конспекта углубляются, расширяются и закрепляются знания, а также дополняется, исправляется и совершенствуется конспект.

Подготовленный конспект и рекомендуемая литература используются при подготовке к семинарским и практическим занятиям. Подготовка сводится к внимательному прочтению учебного материала, к выводу с карандашом в руках всех утверждений и формул, к решению примеров, задач, к ответам на вопросы. Примеры, задачи, вопросы по теме являются средством самоконтроля.

Непременным условием глубокого усвоения учебного материала является знание основ, на которых строится изложение материала. Обычно преподаватель напоминает, какой ранее изученный материал и в какой степени требуется подготовить к очередному занятию. Обращение к ранее изученному материалу не только помогает восстановить в памяти известные положения, выводы, но и приводит разрозненные знания в систему, углубляет и расширяет их. Каждый возврат к старому материалу позволяет найти в нем что-то новое, переосмыслить его с иных позиций, определить для него наиболее подходящее место в уже имеющейся системе знаний. Неоднократное обращение к пройденному материалу является наиболее рациональной формой приобретения и закрепления знаний.

Методические рекомендации студентам по работе с литературой

В рабочей программе дисциплины для каждого раздела и темы дисциплины указывается основная и дополнительная литература, позволяющая более глубоко изучить данный вопрос. Обычно список всей рекомендуемой литературы преподаватель озвучивает на первой лекции или дает ссылки на ее местонахождение (на образовательном портале РГРТУ, на сайте кафедры и т. д.).

При работе с рекомендуемой литературой целесообразно придерживаться такой последовательности. Сначала лучше прочитать заданный текст в быстром темпе. Цель такого чтения заключается в том, чтобы создать общее представление об изучаемом материале, понять общий смысл прочитанного. Затем прочитать вторично, более медленно, чтобы в ходе чтения понять и запомнить смысл каждой фразы, каждого положения и вопроса в целом.

Чтение приносит пользу и становится продуктивным, когда сопровождается записями. Это может быть составление плана прочитанного текста, тезисы или выписки, конспектирование и др. Выбор вида записи зависит от характера изучаемого материала и целей работы с ним. Если содержание материала несложное, легко усваиваемое, можно ограничиться составлением плана. Если материал содержит новую и трудно усваиваемую информацию, целесообразно его законспектировать.

План – это схема прочитанного материала, перечень вопросов, отражающих структуру и последовательность материала.

Конспект – это систематизированное, логичное изложение материала источника. Различаются четыре типа конспектов:

- план-конспект – это развернутый детализированный план, в котором по наиболее сложным вопросам даются подробные пояснения;
- текстуальный конспект – это воспроизведение наиболее важных положений и фактов источника;
- свободный конспект – это четко и кратко изложенные основные положения в результате глубокого изучения материала, могут присутствовать выписки, цитаты, тезисы; часть материала может быть представлена планом;
- тематический конспект – составляется на основе изучения ряда источников и дает ответ по изучаемому вопросу.

В процессе изучения материала источника и составления конспекта нужно обязательно применять различные выделения, подзаголовки, создавая блочную структуру конспекта. Это делает конспект легко воспринимаемым и удобным для работы.

Методические рекомендации студентам по подготовке к практическим занятиям/лабораторным работам

Цель практических занятий – способствовать закреплению теоретических знаний, приобретению и развитию практических умений решать задачи, умений и практических навыков применять общие закономерности к конкретным случаям.

Практические занятия: стимулируют регулярное изучение рекомендованной литературы, а также внимательное отношение к лекционному курсу, закрепляют знания, полученные в процессе лекционного обучения и самостоятельной работы над литературой, расширяют объем профессионально значимых знаний, умений, навыков, позволяют проверить правильность ранее полученных знаний, прививают навыки самостоятельного мышления, устного выступления, способствуют свободному оперированию терминологией, предоставляют преподавателю возможность систематически контролировать уровень самостоятельной работы студентов.

Лабораторная работа – это форма организации учебного процесса, когда обучающиеся по заданию и под руководством преподавателя самостоятельно выполняют специально

разработанные задания.

Для проведения лабораторных работ используется вычислительная техника, которые размещаются в специально оборудованных учебных лабораториях. Перед началом цикла лабораторных работ преподаватель или другое ответственное лицо проводит с обучающимися инструктаж о правилах техники безопасности в данной лаборатории, после чего студенты расписываются в специальном журнале техники безопасности.

По каждой лабораторной работе разрабатываются методические указания по их проведению. Они используются обучающимися при выполнении лабораторной работы.

Применяются разные формы организации обучающихся на лабораторных работах: фронтальная, групповая и индивидуальная. При фронтальной форме организации занятий все обучающиеся выполняют одновременно одну и ту же работу. При групповой форме организации занятий одна и та же работа выполняется группами по 2-5 человек. При индивидуальной форме организации занятий каждый обучающийся выполняет индивидуальное задание. Выбор метода зависит от учебно-методической базы и задач курса.

До начала лабораторной работы студент должен ознакомиться с теоретическими вопросами, которые будут изучаться или исследоваться в этой работе. Перед началом лабораторной работы преподаватель может провести проверку знаний обучающихся – их теоретической готовности к выполнению задания. По итогам этой проверки студент допускается или не допускается к данной работе. О такой исходной проверке преподаватель информирует студентов заранее. Также возможна ситуация, когда допуском к очередной лабораторной работе является своевременная сдача предыдущей лабораторной работы (или подготовка отчета по ней).

Во время лабораторной работы обучающиеся выполняют запланированное лабораторное задание. Все полученные результаты необходимо зафиксировать в черновике отчета или сохранить в электронном виде на сменном носителе.

Завершается лабораторная работа оформлением индивидуального отчета и его защитой перед преподавателем.

Приступая к работе в лаборатории, студенту следует знать, что в отличие от других видов занятий, пропущенную или некачественно выполненную лабораторную работу нельзя отработать в любое время. Для этого существуют специальные дополнительные дни ликвидации учебных задолженностей. Поэтому пропускать лабораторную работу без уважительной причины крайне нежелательно.

Методические рекомендации студентам по подготовке к зачету/экзамену

При подготовке к зачету/экзамену студент должен повторно изучить конспекты лекций и рекомендованную литературу, просмотреть решения основных задач, решенных самостоятельно и на семинарах.

Необходимо помнить, что практически все зачеты и экзамены в вузе сконцентрированы в течение не очень большого временного периода в конце семестра в соответствии с расписанием. Промежутки между очередными зачетами и экзаменами обычно составляют всего несколько дней. Поэтому подготовку к ним нужно начинать заблаговременно в течение семестра. До наступления промежуточной аттестации уточните у преподавателя порядок проведения промежуточной аттестации по его предмету и формулировки критериев для количественной оценивания уровня подготовки студентов. Для итоговой положительной оценки по предмету необходимо вовремя и с нужным качеством выполнить или защитить контрольные работы, типовые расчеты, лабораторные работы и т. д., т. к. всё это может являться обязательной частью учебного процесса по данной дисциплине.

Рекомендуется разработать план подготовки к каждому зачету и экзамену, в котором указать, какие вопросы или билеты нужно выучить, какие задачи решить за указанный в плане

временной отрезок.

Также бывает полезно вначале изучить более сложные вопросы, а затем переходить к изучению более простых вопросов. При этом желательно в начале каждого следующего дня подготовки бегло освежить в памяти выученный ранее материал.

В период сдачи зачета/экзамена организм студента работает в крайне напряженном режиме и для успешной сдачи промежуточной аттестации нужно не забывать о простых, но обязательных правилах:

- по возможности обеспечить достаточную изоляцию: не отвлекаться на разговоры с друзьями, просмотры телепередач, общение в социальных сетях;
- уделять достаточное время сну;
- отказаться от успокоительных, здоровое волнение – это нормально, лучше снимать волнение небольшими прогулками, самовнушением;
- внушать себе, что промежуточная аттестация – это не проблема, это нормальный рабочий процесс, не накручивайте себя, не создавайте трагедий в своей голове;
- помогите своему организму – обеспечьте ему полноценное питание, давайте ему периоды отдыха с переменной вида деятельности;
- следуйте плану подготовки.

Методические рекомендации студентам по проведению самостоятельной работы

Самостоятельная работа студента над учебным материалом является неотъемлемой частью учебного процесса в вузе.

В учебном процессе образовательного учреждения выделяются два вида самостоятельной работы.

1. Аудиторная – выполняется на учебных занятиях, под непосредственным руководством преподавателя и по его заданию), студентам могут быть предложены следующие виды заданий:

- выполнение самостоятельных работ;
- выполнение лабораторных работ/практических заданий;
- составление схем, диаграмм, заполнение таблиц;
- решение задач;
- работу со справочной, нормативной документацией и научной литературой;
- защиту выполненных работ;
- тестирование и т. д.

2. Внеаудиторная – выполняется по заданию преподавателя, но без его непосредственного участия, включает следующие виды деятельности.

- подготовку к аудиторным занятиям (теоретическим и лабораторным работам);
- изучение учебного материала, вынесенного на самостоятельную проработку: работа над определенными темами, разделами, вынесенными на самостоятельное изучение в соответствии с рабочими программами учебной дисциплины или профессионального модуля;
- выполнение домашних заданий разнообразного характера;
- выполнение индивидуальных заданий, направленных на развитие у студентов самостоятельности и инициативы;
- подготовку к лабораторной работе, практическому занятию, зачету, экзамену;
- другие виды внеаудиторной самостоятельной работы.

Внеаудиторные самостоятельные работы представляют собой логическое продолжение аудиторных занятий, проводятся по заданию преподавателя, который инструктирует студентов и устанавливает сроки выполнения задания.

При планировании заданий для внеаудиторной самостоятельной работы используются следующие типы самостоятельной работы:

– воспроизводящая (репродуктивная), предполагающая алгоритмическую деятельность по образцу в аналогичной ситуации. Включает следующую основную деятельность: самостоятельное прочтение, просмотр, конспектирование учебной литературы, прослушивание записанных лекций, заучивание, пересказ, запоминание, Internet–ресурсы, повторение учебного материала и др.;

– реконструктивная, связанная с использованием накопленных знаний и известного способа действия в частично измененной ситуации, предполагает подготовку отчетов по лабораторным работам, подбор литературы по дисциплинарным проблемам, подготовка к защите лабораторных работ и др.;

– эвристическая (частично-поисковая) и творческая, направленная на развитие способностей студентов к исследовательской деятельности.

Одной из важных форм самостоятельной работы студента является работа с литературой ко всем видам занятий.

Один из методов работы с литературой – повторение: прочитанный текст можно заучить наизусть. Простое повторение воздействует на память механически и поверхностно. Полученные таким путем сведения легко забываются.

Более эффективный метод – метод кодирования: прочитанный текст нужно подвергнуть большей, чем простое заучивание, обработке. Чтобы основательно обработать информацию и закодировать ее для хранения, важно провести целый ряд мыслительных операций: прокомментировать новые данные; оценить их значение; поставить вопросы; сопоставить полученные сведения с ранее известными. Для улучшения обработки информации очень важно устанавливать осмысленные связи, структурировать новые сведения.

Изучение научной учебной и иной литературы требует ведения рабочих записей. Форма записей может быть весьма разнообразной: простой или развернутый план, тезисы, цитаты, конспект.

План – структура письменной работы, определяющая последовательность изложения материала. Он является наиболее краткой и потому самой доступной и распространенной формой записей содержания исходного источника информации. По существу, это перечень основных вопросов, рассматриваемых в источнике. План может быть простым и развернутым. Их отличие состоит в степени детализации содержания и, соответственно, в объеме.

Преимущество плана состоит в том, что план позволяет наилучшим образом уяснить логику мысли автора, упрощает понимание главных моментов произведения. Кроме того, он позволяет быстро и глубоко проникнуть в сущность построения произведения и, следовательно, гораздо легче ориентироваться в его содержании и быстрее обычного вспомнить прочитанное. С помощью плана гораздо удобнее отыскивать в источнике нужные места, факты, цитаты и т. д.

Выписки представляют собой небольшие фрагменты текста (неполные и полные предложения, отдельные абзацы, а также дословные и близкие к дословным записи об излагаемых в нем фактах), содержащие в себе квинтэссенцию содержания прочитанного. Выписки представляют собой более сложную форму записи содержания исходного источника информации. По сути, выписки – не что иное, как цитаты, заимствованные из текста. Выписки позволяют в концентрированной форме и с максимальной точностью воспроизвести наиболее важные мысли автора. В отдельных случаях – когда это оправдано с точки зрения продолжения работы над текстом – вполне допустимо заменять цитирование изложением, близким дословному.

Тезисы – сжатое изложение содержания изученного материала в утвердительной (реже опровергающей) форме. Отличие тезисов от обычных выписок состоит в том, что тезисам присуща значительно более высокая степень концентрации материала. В тезисах отмечается преобладание выводов над общими рассуждениями. Записываются они близко к оригинальному тексту, т. е. без использования прямого цитирования.

Аннотация – краткое изложение основного содержания исходного источника информации, дающее о нем обобщенное представление. К написанию аннотаций прибегают в тех случаях, когда подлинная ценность и пригодность исходного источника информации исполнителю письменной работы окончательно неясна, но в то же время о нем необходимо оставить краткую запись с обобщающей характеристикой.

Резюме – краткая оценка изученного содержания исходного источника информации, полученная, прежде всего, на основе содержащихся в нем выводов. Резюме весьма сходно по своей сути с аннотацией. Однако, в отличие от последней, текст резюме концентрирует в себе данные не из основного содержания исходного источника информации, а из его заключительной части, прежде всего выводов. Но, как и в случае с аннотацией, резюме излагается своими словами – выдержки из оригинального текста в нем практически не встречаются.

Конспект представляет собой сложную запись содержания исходного текста, включающая в себя заимствования (цитаты) наиболее примечательных мест в сочетании с планом источника, а также сжатый анализ записанного материала и выводы по нему.

При выполнении конспекта требуется внимательно прочитать текст, уточнить в справочной литературе непонятные слова и вынести справочные данные на поля конспекта. Нужно выделить главное, составить план. Затем следует кратко сформулировать основные положения текста, отметить аргументацию автора. Записи материала следует проводить, четко следуя пунктам плана и выражая мысль своими словами. Цитаты должны быть записаны грамотно, учитывать лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля. Необходимо указывать библиографическое описание конспектируемого источника.

Методические рекомендации студентам по выполнению курсовой работы/курсового проекта

Курсовая работа/курсовой проект представляет собой самостоятельное законченное теоретическое и (или) прикладное исследование на заданную (выбранную) тему, написанное студентом под руководством преподавателя, свидетельствующее об умении студента работать с литературой, обобщать и анализировать фактический материал, используя теоретические знания и практические навыки, полученные при освоении профессиональной образовательной программы.

Для дисциплины учебным планом определена требуемая форма контроля: курсовая работа или курсовой проект. Основное отличие их заключается в том, что в курсовом проекте, в отличие от курсовой работы, обязательно должна быть выделенная практическая часть, связанная с выполнением достаточно сложной расчетной и (или) проектной задачей.

Выполнение студентом курсовой работы/курсового проекта осуществляется в процессе изучения учебной дисциплины, а защита происходит на заключительном этапе изучения дисциплины.

Темы курсовых работ/курсовых проектов определяются преподавателем дисциплины. Тема студенту может быть назначена преподавателем, а может быть выбрана студентом самостоятельно из предложенного списка.

Оформление работы должно соответствовать требованиям, изложенным в соответствующих

методических указаниях к курсовой работе/курсовому проекту, которые должен предоставить студентам преподаватель.

Курсовая работа/курсовой проект не является пересказом изученного материала, а представляет его творческую переработку, что предполагает самостоятельное исследование студентом проблематики избранной темы. Курсовая работа/курсовой проект не может состоять из фрагментов статей и публикаций, напоминая школьное сочинение. Будучи учебным исследованием, данный вид работы должен по своему содержанию и форме стремиться к стандартам научного текста.

Для успешного и качественного выполнения курсовой работы/курсового проекта студенту необходимо:

- использовать методы научного исследования;
- ориентироваться в различных источниках информации и правильно работать со специальной литературой;
- уметь грамотно и научно обоснованно формулировать теоретические рекомендации, результаты анализа;
- квалифицированно оформлять графический материал, иллюстрирующий содержание курсовой работы/курсового проекта.

Курсовая работа/курсовой проект имеет свою структуру, сохранение которой обязательно. Она включает следующие элементы:

- титульный лист;
- задание;
- содержание;
- введение
- основная часть;
- заключение
- список использованных источников
- приложения (при необходимости).

В установленные преподавателем сроки студент должен предоставить сброшюрованную бумажную курсовую работу/курсовой проект. Далее студент в установленные преподавателем сроки должен пройти защиту курсовой работы/курсового проекта. Курсовая работа/курсовой проект с рецензией преподавателя должны быть размещены в портфолио студента в ЭИОС.