

**Автономная некоммерческая организация профессионального образования  
«ПЕРМСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»  
(АНО ПО «ПГТК»)**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ  
ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ ПО  
ДИСЦИПЛИНЕ  
ОП.08 УПРАВЛЕНИЕ IT-ПРОЕКТАМИ  
для специальности  
09.02.11 Разработка и управление программным обеспечением  
(код и наименование специальности)**

**Квалификация выпускника  
Программист**

**Форма обучения  
Очная**

Пермь 2026

Методические рекомендации по выполнению практических работ учебной дисциплины ОП.08 УПРАВЛЕНИЕ ИТ-ПРОЕКТАМИ составлен в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением

Данные методические рекомендации помогут организовать самостоятельную деятельность студентов на основе деятельного и компетентного подходов к обучению, что соответствует ФГОС СПО по специальности 09.02.11 Разработка и управление программным обеспечением.

Автор – составитель: Могильникова Н.С., старший преподаватель.

Методические рекомендации по выполнению практических работ предназначен для оценивания достижений запланированных результатов по дисциплине ОП.08 УПРАВЛЕНИЕ ИТ-ПРОЕКТАМИ. Методические рекомендации по выполнению практических работ представляет собой комплект материалов для проведения практических занятий (в форме практической подготовке) и осуществления контроля за выполнением работ.

Методические рекомендации по выполнению практических работ позволяет оценивать:

| Код ОК, ПК  | Уметь  | Знать   |
|---|--|---|
| ОК 01 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.<br>ОК 02 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;<br>ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках. | <ul style="list-style-type: none"> <li>• применять на практике принципы и практику управления коммуникациями проекта;</li> <li>• планировать и управлять сроками;</li> <li>• выявлять и уменьшать риски;</li> <li>• оценивать сложность поддержки проекта и связанные с этим изменения его стоимости;</li> <li>• оценивать проектные решения, характер тенденций и экономические характеристики проекта;</li> <li>• находить баланс между квалификацией персонала, затратами на его обучение, качеством продукта и соблюдением сроков;</li> <li>• работать индивидуально и в команде в качестве члена группы по планированию и управлению процессами создания аппаратно-программных средств</li> </ul> | <ul style="list-style-type: none"> <li>• основные понятия, принципы и практики управления проектами, включая методы планирования проектов;</li> <li>• модели жизненного цикла проекта;</li> <li>• методологию управления программными проектами;</li> <li>• инструменты проектирования, планирования и оперативного управления;</li> <li>• методы контроля качества;</li> <li>• методологии построения команды</li> </ul> |

### Практическое занятие «Использование метрик программного продукта»

Целью работы является изучение метрик стилистики компьютерных программ.

Результатом практической работы является отчет, в котором должны быть приведены метрические параметры стилистики.

Для выполнения практической работы студент должен изучить приведенный ниже теоретический материал. Для вычисления параметров метрики потока данных программ необходимо использовать формулы расчета для метрики стилистики программ.

Отчет сдается в распечатанном и электронном (файл Word) видах.

Метрики стилистики программ

За время своего существования программирование перестало быть искусством отдельных исполнителей, превратившись в предмет коллективной производственной

деятельности. Язык программирования и сами программы являются не только средством общения программиста с компьютером, но и средством общения программистов между собой. Появляются новые требования к читаемости и воспринимаемости программ, соблюдение которых упрощает контакт внутри группы разработчиков, а также позволяет обслуживать и корректировать программы без участия непосредственных разработчиков.

Наиболее простой метрикой стилистики и понятности является оценка уровня комментированности программы  $F$ :

$$F = N_{\text{ком}}/N_{\text{стр}},$$

где  $N_{\text{ком}}$  – количество комментариев в программе;  $N_{\text{стр}}$  – количество строк или операторов исходного текста.

Таким образом, метрика  $F$  отражает насыщенность программы комментариями.

Исходя из практического опыта принято считать, что  $F > 0.1$ , т.е. на каждые десять строк программы должен приходиться минимум один комментарий. Как показывают исследования, комментарии распределяются по тексту программы неравномерно: в начале программы их избыток, а в середине или в конце – недостаток. Это объясняется тем, что в начале программы, как правило, расположены операторы описания идентификаторов, требующие более «плотного» комментирования. Кроме того, в начале программы также расположены «шапки», содержащие общие сведения об исполнителе, характере, функциональном назначении программы и т.п.

Такая насыщенность компенсирует недостаток комментариев в теле программы, и поэтому приведенная формула недостаточно точно отражает комментированность функциональной части текста программы.

Более удачен вариант, когда вся программа разбивается на  $n$  равных сегментов и для каждого из них определяется  $F_i$

Уровень комментированности программы считается нормальным, если выполняется условие:  $F = n$ . В противном случае какой-либо фрагмент программы дополняется комментариями до нормального уровня.

Необходимо подчеркнуть, что стилистика и понятность программ тесно связана и с размером и со сложностью программ. Поэтому не следует забывать о некоторой условности группировки метрик программ.

Контрольные вопросы

- 1 Для чего рассчитываются метрики стилистики?
- 2 Какие метрики стилистики существуют?
- 3 Как рассчитывается оценка уровня комментированности программы?
- 4 Какой уровень комментированности считается нормальным?

### **Практическое занятие «Проверка целостности программного кода»**

Целью работы — изучение методов защиты программного кода от злоумышленников, а также проектирование и разработка своего метода защиты ПО. Создание своего метода защиты ПО.

Создать программу в среде Visual Studio 2019 на языке программирования C++.

В процессе занятия были поставлены и последовательно решены следующие задачи:

- 1 Узнать о методах защиты программного кода от обратного проектирования.
- 2 Разобраться, какой из них лучше всего применять на практике (или же стоит применять их комбинации).

3 Дать общую оценку методу защиты.

4 Разработать свой метод защиты ПО.

### **Практическое занятие «Анализ потоков данных»**

Цель работы: Ознакомление пользователя с методологией графического структурного анализа и существующими структурами организации потоков данных, получение практических навыков в работе с Data Flow Diagrams.

#### **ЗАДАНИЕ**

Смоделировать работу системы технического обслуживания воздушного судна по указанию преподавателя.

#### **Контрольные вопросы**

1. Что такое DFD?
2. Какие новые типы объектов может содержать DFD?
3. Какие физические характеристики системы отражаются в DFD-моделях?
4. Область применения DFD?
5. Привести пример контекстной DFD-диаграммы?
6. Назначение функциональных блоков, внешних сущностей и стрелок (потоков данных) в DFD?
7. Какова функция хранилищ данных в производственных и информационных системах?
8. Привести пример разветвления и объединения стрелок?
9. Каковы основные подходы к построению DFD-моделей?
10. Принципы нумерации объектов в DFD?

### **Практическое занятие «Использование метрик стилистики»**

Целью работы является изучение метрик потока данных компьютерных программ. Результатом практической работы является отчет, в котором должны быть приведены метрические параметры потока данных программ.

Для выполнения практической работы студент должен изучить приведенный ниже теоретический материал. Для вычисления параметров метрики потока данных программ необходимо использовать формулы расчета для метрики сложности программ. Отчет сдается в распечатанном и электронном (файл Word) видах.

#### **Метрики сложности программ**

При оценке сложности программ, как правило, выделяют три основные группы метрик: метрики размера программ, метрики сложности потока управления программ и метрики сложности потока данных программ.

Оценки первой группы наиболее просты и поэтому получили широкое распространение. Традиционной характеристикой размера программ является количество строк исходного текста. Под строкой понимается любой оператор программы.

Непосредственное измерение размера программы, несмотря на свою простоту, дает хорошие результаты. Оценка размера программы недостаточна для принятия решения о ее сложности. Но вполне применима для классификации программ, существенно различающихся объемами. При уменьшении различий в объеме программ на первый план выдвигаются оценки других факторов, оказывающих влияние на сложность. Таким образом, оценка размера программы есть оценка по номинальной шкале, на основе которой определяются только категории программ без уточнения оценки для каждой категории.

К группе оценок размера программ можно отнести также метрику Холстеда. За базу принят подсчет количества операторов и операндов используемых в программе, т.е. определение размера программы.

Основу метрики Холстеда составляют четыре измеряемые характеристики программы:  $h_1$  - число уникальных операторов программы, включая символы-разделители, имена процедур и знаки операций (словарь операторов);  $h_2$  – число уникальных операндов программы (словарь операндов);  $N_1$  – общее число операторов в программе  $N_2$  – общее число операндов в программе.

Опираясь на эти характеристики, получаемые непосредственно при анализе исходных текстов программ, Холстед вводит следующие оценки

Словарь программы  $h = h_1 + h_2$

Длину программы  $N = N_1 + N_2$

Объем программы  $V = N \log_2 h$

Смысл оценок  $h$  и  $N$  достаточно очевиден, поэтому подробно рассмотрим только характеристику  $V$ .

Количество символов, используемых при реализации некоторого алгоритма, определяется в числе прочих параметров и словарей программы  $h$ , представляющим собой минимально необходимое число символов, обеспечивающих реализацию алгоритма.

Далее Холстед вводит  $h^*$  - теоретический словарь программы, т.е. словарный запас, необходимый для написания программы с учетом того, что необходимая функция уже реализована в данном языке и, следовательно, программа сводится к вызову этой функции. Например, согласно Холстеду возможное осуществление процедуры выделения простого числа могло бы выглядеть так: CALL SIMPLE (X, Y),

где Y- массив численных значений, содержащих искомое число X.

Теоретический словарь в этом случае будет состоять из

$n_1^*$ : {CALL, SIMPLE (...)}  $n_1^* = 2$ ;

$n_2^*$ : {X, Y},  $h_2^* = 2$ ;

а его длина, определяемая как

$h^* = h_1^* + h_2^*$  будет равна 4

Используя  $h^*$ , Холстед вводит оценку  $V^*$ :  $V^* = h^* \log_2 h^*$ , с помощью которой описывается потенциальный объем программы, соответствующий максимально компактно реализующей данный алгоритм.

Другая группа метрик сложности программ – метрика сложности потока данных, то есть использования, конфигурации и размещения данных в программах.

Пара “модуль – глобальная переменная” обозначается как (p,r), где p – модуль, имеющий доступ к глобальной переменной r. В зависимости от наличия в программе реального обращения к переменной r формируются два типа пар «модуль – глобальная переменная»: фактические и возможные. Возможное обращение к r с помощью p показывает, что область существования r включает в себя p.

Характеристика  $A_{up}$  говорит о том, сколько раз модули  $U_p$  действительно получили доступ к глобальным переменным, а число  $P_{up}$  – сколько раз они могли бы получить доступ.

Отношение числа фактических обращений к возможным определяется

$R_{up} = A_{up} / P_{up}$

Эта формула показывает приближенную вероятность ссылки произвольного модуля на произвольную глобальную переменную. Очевидно, чем выше эта вероятность, тем выше вероятность “несанкционированного” изменения какой-либо переменной, что может

существенно усложнить работы, связанные с модификацией программы.

Покажем расчет метрики “модуль – глобальная переменная”. Пусть в программе имеются три глобальные переменные и три подпрограммы. Если предположить, что каждая подпрограмма имеет доступ к каждой из переменных, то мы получим девять возможных пар, то есть  $R_{up}=9$ . Далее пусть первая подпрограмма обращается к одной переменной, вторая – двум, а третья не обращается ни к одной переменной. Тогда  $A_{up}=3$ ,  $R_{up}=3/9$ .

Еще одна метрика сложности потока данных – спен.

Определение спена основывается на локализации обращения к данным внутри каждой программной секции.

Спен – это число утверждений, содержащих данный идентификатор, между его первым и последним появлением в тексте программы. Идентификатор, появившийся  $n$  раз, имеет спен, равный  $n$ . Спен определяет количество контролируемых утверждений, вводимых в тело программы при построении трассы программы по этому идентификатору в процессе тестирования и отладки.

Следующей метрикой сложности потока данных программ является метрика Чепина. Существует несколько ее модификаций. Рассмотрим более простой, а с точки зрения практического использования – достаточно эффективный вариант этой метрики.

Суть метода состоит в оценке информационной прочности отдельно взятого программного модуля с помощью анализа характера использования переменных из списка ввода-вывода.

Все множество переменных, составляющих список ввода-вывода, разбивается на четыре функциональные группы

$P$  – вводимые переменные для расчетов и для обеспечения вывода. Примером может служить используемая в программах лексического анализатора переменная, содержащая строку исходного текста программы, то есть сама переменная не модифицируется, а только содержит исходную информацию.

$M$  – модифицируемые или создаваемые внутри программы переменные.

$C$  – переменные, участвующие в управлении работой программного модуля (управляющие переменные).

Не используемые в программе («паразитные») переменные. Поскольку каждая переменная может выполнять одновременно несколько функций, необходимо учитывать ее в каждой соответствующей функциональной группе.

Далее вводится значение метрики Чепина:

$$Q = a_1P + a_2M + a_3C + a_4T,$$

где  $a_1, a_2, a_3, a_4$  – весовые коэффициенты.

Весовые коэффициенты использованы для отражения различного влияния на сложность программы каждой функциональной группы. По мнению автора метрики, наибольший вес, равный трем, имеет функциональная группа  $C$ , так как она влияет на поток управления программой. Весовые коэффициенты остальных групп распределяются следующим образом:  $a_1=1$ ;  $a_2=2$ ;  $a_4=0.5$ . Весовой коэффициент группы  $T$  не равен нулю, поскольку «паразитные» переменные не увеличивают сложности потока данных программы, но иногда затрудняют ее понимание. С учетом весовых коэффициентов выражение примет вид

$$Q = P + 2M + 3C + 0.5T.$$

Следует отметить, что рассмотренные метрики сложности программы основаны на анализе исходных текстов программ, что обеспечивает единый подход к автоматизации их

расчета. Контрольные вопросы

- 1 Какие метрики оценки сложности программ существуют?
- 2 Какие характеристики составляют метрику Холстеда?
- 3 Что такое метрика сложности потока данных?
- 4 Что такое спен?
- 5 Как рассчитывается метрика Чепина?

### **Практическое занятие «Выполнение измерений характеристик кода в среде VisualStudio»**

Цель работы: изучить среду быстрой разработки приложений Visual Studio. Научиться размещать и настраивать внешний вид элементов управления на форме.

По указанию преподавателя выберите свое индивидуальное задание. Уточните условие задания, количество, наименование, типы исходных данных. Прочтите описание свойств и описание элементов управления Form, Label, TextBox, Button. С помощью окна свойств установите первоначальный цвет формы, шрифт выводимых символов.

#### **Индивидуальные задания**

1 Разместите на форме четыре кнопки (Button). Сделайте на кнопках следующие надписи: «красный», «зеленый», «синий», «желтый». Создайте четыре обработчика события нажатия на данные кнопки, которые будут менять цвет формы в соответствии с текстом на кнопках.

2 Разместите на форме две кнопки (Button) и одну метку (Label). Сделайте на кнопках следующие надписи: «привет», «до свидания». Создайте обработчики события нажатия на данные кнопки, которые будут менять текст метки на слова, написанные на кнопках. Создайте обработчик события создания формы (Load), который будет устанавливать цвет формы и менять текст метки на строку «Начало работы».

3 Разместите на форме ряд кнопок (Button) напротив каждой поле ввода (TextBox) и одну метку (Label). Создайте обработчики события нажатия на данные кнопки, которые будут менять текст в метке. Текст в метке берется из поля ввода напротив нажимаемой кнопки.

4 Разместите на форме ряд кнопок (Button), и одно поле ввода (TextBox). Создайте обработчики события нажатия на данные кнопки, которые будут менять текст на нажатой кнопке. Текст на кнопке берется из поля ввода.

5 Разместите на форме ряд кнопок (Button) и ряд меток (Label). Создайте обработчики события нажатия на данные кнопки, которые будут менять цвет двух меток. Создайте обработчик события нажатия кнопки мыши на форме (Click), который будет устанавливать цвет всех меток в белый.

6 Разместите на форме ряд кнопок (Button) и ряд меток (Label). Создайте обработчик события создания формы (Load), который будет делать все метки невидимыми. Создайте обработчики события нажатия на кнопки, которые будут менять свойство метки Visible, тем самым делать их видимыми.

7 Разместите на форме ряд кнопок (Button), напротив каждой поле ввода (TextBox). Создайте обработчики события нажатия на данные кнопки, которые будут менять заголовок окна. Текст в заголовке берет- ся из поля ввода напротив нажимаемой кнопки.

8 Разместите на форме две кнопки (Button) и одну метку (Label). Сделайте на кнопках следующие надписи: «скрыть», «показать». Создайте обработчики события нажатия на данные кнопки, которые будут скрывать или показывать метку. Создайте обработчик



события создания формы (Load), который будет устанавливать цвет формы и менять текст метки на строку «Начало работы».

9 Разместите на форме три кнопки (Button) и одно поле ввода (TextBox). Сделайте на кнопках следующие надписи: «скрыть», «показать», «очистить». Создайте обработчики события нажатия на данные кнопки, которые будут скрывать или показывать поле ввода. При нажатии на кнопку «очистить» текст из поля ввода должен быть удален.

10 Разместите на форме две кнопки (Button) и одно поле ввода (TextBox). Сделайте на кнопках следующие надписи: «заполнить», «очистить». Создайте обработчики события нажатия на данные кнопки, которые будут очищать или заполнять поле ввода знаками «\*\*\*\*\*». Создайте обработчик события создания формы (Load), который будет устанавливать цвет формы и менять текст в поле ввода на строку «+++++».

11 Разработайте игру, которая заключается в следующем. На форме размещены пять кнопок (Button). При нажатии на кнопку некоторые кнопки становятся видимыми, а другие – невидимыми. Цель игры – скрыть все кнопки.

12 Разработайте игру, которая заключается в следующем. На форме размещены четыре кнопки (Button) и четыре метки (Label). При нажатии на кнопку часть надписей становится невидимой, а часть, наоборот, становится видимой. Цель игры – скрыть все надписи.

13 Разместите на форме ряд кнопок (Button). Создайте обработчики события нажатия на данные кнопки, которые будут делать неактивными текущую кнопку. Создайте обработчик события изменения размера формы (Resize), который будет устанавливать все кнопки в активный режим.

14 Разместите на форме ряд кнопок (Button). Создайте обработчики события нажатия на данные кнопки, которые будут делать неактивными следующую кнопку. Создайте обработчик события нажатия кнопки мыши на форме (Click), который будет устанавливать все кнопки в активный режим.

15 Разместите на форме три кнопки (Button) и одно поле ввода (TextBox). Сделайте на кнопках следующие надписи: «\*\*\*\*\*», «+++++», «00000». Создайте обработчики события нажатия на данные кнопки, которые будут выводить текст, написанный на кнопках, в поле ввода. Создайте обработчик события создания формы (Load), который будет устанавливать цвет формы и менять текст в поле ввода на строку «Готов к работе».

16 Разместите на форме ряд полей ввода (TextBox). Создайте обработчики события нажатия кнопкой мыши на данные поля ввода, которые будут выводить в текущее поле ввода его номер. Создайте обработчик события изменения размера формы (Resize), который будет очищать все поля ввода.

17 Разместите на форме поле ввода (TextBox), метку (Label) и кнопку (Button). Создайте обработчик события нажатия на кнопку, который будет копировать текст из поля ввода в метку. Создайте обработчик события нажатия кнопки мышки на форме (Click), который будет устанавливать цвет формы и менять текст метки на строку «Начало работы» и очищать поле ввода.

18 Разместите на форме поле ввода (TextBox) и две кнопки (Button) с надписями: «блокировать», «разблокировать». Создайте обработчики события нажатия на кнопки, которые будут делать активным или неактивным поле ввода. Создайте обработчик события нажатия кнопки мышки на форме (Click), который будет устанавливать цвет формы и делать невидимыми все элементы.

19 Реализуйте игру минер на поле  $3 \times 3$  из кнопок (Button). Первоначально все кнопки

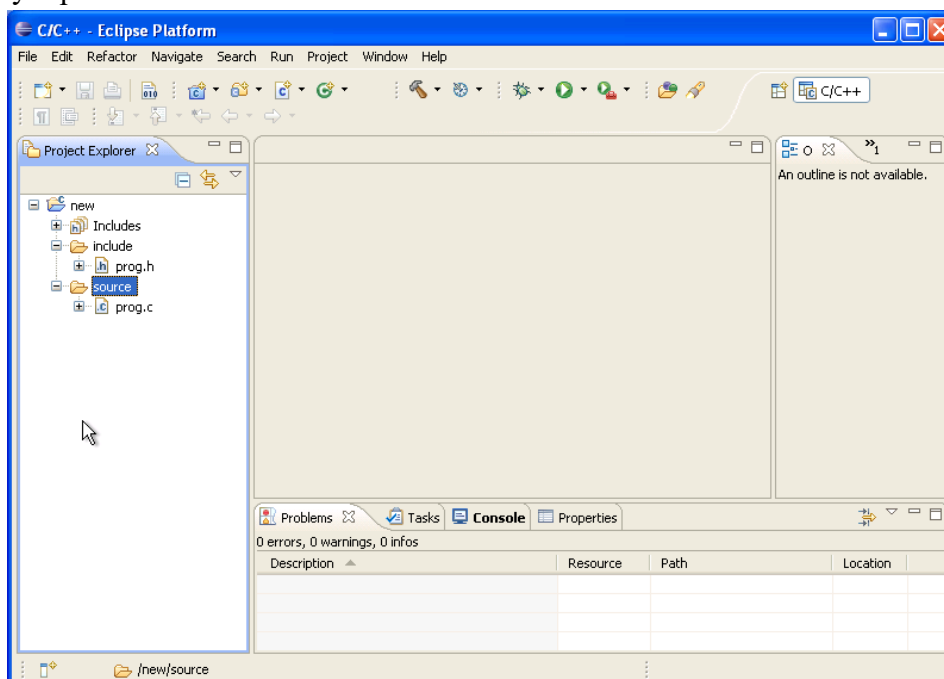
не содержат надписей. При попытке нажатия на кнопку на ней либо показывается количество мин, либо надпись «ми- на!» и меняется цвет окна.

20 Разместите на форме четыре кнопки (Button). Напишите для каждой обработчик события, который будет менять размеры и местоположение на окне других кнопок.

### **Практическое занятие «Выполнение измерений характеристик кода в среде (например, Eclipse C/C++ и др.)»**

Рассмотрим процесс создания проекта на языке C в среде Eclipse. Пусть у нас уже имеются файлы с исходным кодом программы.

1. Запустите среду Eclipse (eclipse.exe).
2. Выберите рабочее пространство "workspace" (если eclipse предлагает сделать это).
3. Создайте новый C-проект : File -> New -> C Project
4. Укажите имя проекта (new) и нажмите finish.
5. Закройте окно Welcome
6. Импортируйте в проект исходные файлы программы. Здесь желательно создать структуру каталогов, в папку include помещаются h-файлы, в папку source - c-файлы. Для добавления каталога нажмите правой кнопкой мыши на имени проекта в окне Project Explorer. Далее New->Folder и указываем имя папки (например, include). Импорт исходного файла в проект: Project Explorer -> Import -> Filesystem. Далее выбираем каталог, в котором располагаются нужные файлы и отмечаем галочками те, которые будут добавлены в проект. Указанные файлы копируются в каталог проекта. Указываем каталог в проекте, в который нужно поместить импортируемые файлы. По завершении получаем структуру проекта, подобную рис. ниже.



Теперь необходимо поместить в проект make-файл. Make-файл указывает компилятору как собрать программу. Незаменим в кроссплатформенных проектах, поскольку позволяет задавать различные варианты сборки для различных платформ (задавать компилятор, степень оптимизации дополнительные библиотеки и пр.). Make-файлы используют в больших проектах, поскольку позволяет повторно компилировать только измененные файлы, что значительно ускоряет процесс сборки программы. В среде Eclipse использование make-файлов является общепринятым методом сборки программы.

Make-файл по умолчанию имеет название "makefile". Он содержит в себе инструкции, описывающие процесс сборки программы. Перевод официального руководства по написанию make-файлов можно взять здесь, детальное описание команд make-файла на примере можно найти здесь. Мы будем использовать простой make-файл, расположенный здесь.

7. Импортируйте в корневой каталог проекта (new) данный make-файл.

Разобраться в структуре make-файла не сложно. Константа CLINKER\_WINDOWS Указывает на путь к компилятору (файлу gcc.exe). В данном случае это "C:\Program files\MinGW\bin\gcc". Измените данную константу, если в вашей системе путь к файлу gcc.exe иной.

Константы source\_dirs и include\_dirs указывают на директории, где компилятор будет искать файлы с исходными текстами и заголовочные файлы соответственно. В данном случае, с-файлы можно располагать в рабочем каталоге и в подкаталоге source. h-файлы можно располагать в рабочем каталоге и в подкаталоге include (точка в списке каталогов означает рабочий каталог программы, список каталогов указывается через пробел, если необходимо указать каталог, содержащий пробелы, то используются двойные кавычки "").

8. Настройте компилятор: Project -> Properties -> Builders.

- Отключите имеющиеся в списке компиляторы (снимите галочку напротив CDT builder и Scanner Configuration Builder).

- Создайте новый компилятор (кнопка new).

- Установите опции компилятора следующим образом:

Вкладка Main:

Location: путь к make-файлу (например, C:\MinGW\bin\mingw32-make.exe).

Working directory: рабочий каталог (например, \${workspace\_loc:/new} - указатель на каталог проекта new)

Arguments: правила компиляции, заданные в make-файле (например, all clean)

Теперь можно проверить правильность настройки компилятора, выбрав команду Project -> Build all или нажать Cntrl+B. Как вы можете видеть, в Eclipse одновременно можно подключать несколько компиляторов.

9. Многие программы требуют специальных параметров для запуска или специального окружения. В Eclipse нельзя просто нажать F9 и запустить скомпилированный \*.exe-файл. Сначала нужно настроить конфигурацию запуска программы. Для этого нужно настроить конфигурацию запуска: Run -> External Tools -> Open External Tools Dialog... и создать новую конфигурацию (New\_configuration). Далее нужно установить опции следующим образом:

Вкладка Main:

Location: путь к программе запуска. В нашем случае это наш скомпилированный exe-файл (например, C:\Documents and settings\andrey\workspace\new\all.exe).

Working directory: рабочий каталог (например, \${workspace\_loc:/new} - указатель на каталог проекта new).

Arguments: необходимые для запуска программы параметры (в нашем случае - ничего).

Теперь можно проверить правильность настройки запуска программы, выбрав команду Run -> External Tools -> New configuration. Если программа отработала успешно, настройка выполнена правильно.

Для дополнительного удобства можно настроить горячие клавиши таким образом, чтобы данная конфигурация запускалась при нажатии клавиши F9.

## **Порядок оформления видов и форм отчетности по работе обучающихся**

### **1. Методические рекомендации по написанию реферата**

Реферат – первая и наиболее частая для студентов с первых курсов обучения форма работы, которая наилучшим образом, обогащает знания и развивает аналитические способности, т.е. способствует формированию профессиональных компетенций, а в воспитательном плане – формирует ответственность и сочетание личного интереса с общественной необходимостью, то есть качества необходимые для будущего специалиста.

#### **Структура работы**

Структура работы, соотношение объема работ по главам в каждом конкретном случае определяются в зависимости от темы, объекта, предмета и целевой направленности исследования.

Типовая структура включает следующие разделы:

1. Титульный лист.
2. Оглавление.
3. Введение.
4. Главы основной части.
5. Заключение.
6. Список используемой литературы и информационных источников.
7. Приложения.

Титульный лист - первая страница работы (на данной странице номер не ставится).

Оглавление - помещается после титульного листа, в нем приводятся пункты работы с указанием страниц (на данной странице номер не ставится).

Введение - кратко обосновывается актуальность выбранной темы, цель и содержание поставленных задач, формулируется объект и предмет исследования, указывается избранный метод исследования. Дается характеристика работы – относится ли она к теоретическим исследованиям или к прикладным, сообщается, в чем заключается значимость и прикладная ценность полученных результатов, приводится характеристика источников для написания работы и краткий обзор имеющейся по данной теме литературы.

Основная часть - подробно приводится методика и техника исследования, даются сведения об объеме исследования, излагаются и обсуждаются полученные результаты. Содержание основной части должно точно соответствовать теме работы и полностью ее раскрывать.

Заключение - содержит основные выводы, к которым автор пришел, в процессе анализа материала (при этом должна быть подчеркнута самостоятельность, новизна, теоретическое и практическое значение полученных результатов).

Список используемой литературы и информационных источников - приводится в конце работы, в алфавитном порядке сначала указываются источники используемой литературы, затем интернет-источники. Допускается использовать в списке литературы источники не позднее 5-летней давности.

Приложение - помещают вспомогательные или дополнительные материалы. В случае необходимости можно привести дополнительные таблицы, рисунки, графики и т.д., если они помогут лучшему пониманию полученных результатов.

#### **Требования к оформлению работы**

Объем работы должен быть 20-30 страниц.

Текст набирается в текстовом редакторе MS Word: шрифт TimesNewRoman, размер – 14 пт, цвет шрифта черный, междустрочный интервал – полуторный (или 1,15), отступ первой строки (абзацный отступ) – 1,25 см, выравнивание текста – по ширине, расстановка

переносов по тексту – автоматическая, в режиме качественной печати. Оглавление должно быть сформировано автоматически. Текст распечатывается на принтере.

Заголовки разделов печатаются строчными буквами с абзацного отступа.

Заголовки подразделов печатаются строчными буквами (кроме первой прописной), располагаются с абзацного отступа. Заголовки пунктов печатаются строчными буквами (кроме первой прописной), с использованием шрифтового выделения (полужирный шрифт), начиная с абзаца. Если заголовок состоит из двух или более предложений, их разделяют точкой. Заголовки подпунктов печатают строчными буквами (кроме первой прописной), начиная с абзаца в подбор к тексту.

В конце заголовков структурных частей, наименований разделов и подразделов точка не ставится. Расстояние между заголовком структурной части (за исключением заголовка пункта) и подразделом должно быть равно 2 интервалам.

Разделы должны иметь порядковую нумерацию в пределах всего текста. Номер раздела указывается перед его названием, после номера раздела точка не ставится, перед заголовком оставляют пробел. Наименования разделов печатаются строчными буквами с абзацного отступа, выделяются полужирным шрифтом размером 16 пт, точка в конце наименования раздела не ставится. Разделы работы оформляются, начиная с новой страницы.

Иллюстрации обозначают словом «Рисунок» и нумеруют последовательно в пределах раздела реферата или сквозной нумерацией. Номер иллюстрации может состоять из номера раздела и порядкового номера иллюстрации, разделенных точкой. Например: «Рисунок 1.2» (второй рисунок первого раздела). Номер иллюстрации, ее название и поясняющие подписи помещают последовательно под иллюстрацией. Если в работе приведена одна иллюстрация, то ее не нумеруют и слово «Рисунок» не пишут. Иллюстрации должны иметь наименование, которое дается после номера рисунка. Точка после номера рисунка и наименования иллюстрации не ставится.

Каждая таблица должна иметь название, которое следует помещать над таблицей слева, без абзацного отступа в одну строку с ее номером через тире. Расстояние от текста до таблицы и от таблицы до последующего текста равно одной строке. Между наименованием таблицы и самой таблицей не должно быть пустых строк.

*Пример.*

Таблица (номер) – Название таблицы

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |

Приложения оформляют как продолжение реферата на последующих страницах, располагая их в порядке появления ссылок в тексте.

Каждое приложение следует начинать с нового листа (страницы) с указанием наверху справа страницы слова «Приложение», напечатанного строчными буквами. Приложение должно иметь содержательный заголовок, расположенный в следующей строке по центру. Если в реферате более одного приложения, их нумеруют последовательно цифрами, например, Приложение 1, Приложение 2 и т.д.

Критерии оценки реферата

К общим критериям можно отнести:

соответствие реферата теме.

глубина и полнота раскрытия темы.

адекватность передачи первоисточника.

логичность, связность.

доказательность.

структурная упорядоченность (наличие введения, основной части, заключения, их оптимальное соотношение).

оформление (наличие оглавления, списка литературы, культура цитирования, сноски и т.д.).

языковая правильность.

Общая оценка за реферат выставляется следующим образом: если студент выполнил от 65% до 80% указанных выше требований, ему ставится минимальный балл – 3 (удовлетворительно); 80-90% — средний балл – 4 (хорошо); 90-100% — максимальный балл – 5 (отлично).

## 2. Методические рекомендации по созданию презентации

В оформлении презентаций выделяют два блока: оформление слайдов и представление информации на них. Для создания качественной презентации необходимо соблюдать ряд требований, предъявляемых к оформлению данных блоков.

Оформление слайдов:

|                      |   |
|----------------------|---|
| Стиль                | <ul style="list-style-type: none"><li>- Соблюдайте единый стиль оформления</li><li>- Избегайте стилей, которые будут отвлекать от самой презентации.</li><li>- Управляющие кнопки не должны преобладать над основной информацией (текстом, иллюстрациями).</li><li>- Для фона и заголовка используйте контрастные цвета. Не используйте на одном слайде больше трех цветов.</li></ul> |
| Фон и цвет           | <ul style="list-style-type: none"><li>- Для фона предпочтительны холодные тона</li><li>- На одном слайде рекомендуется использовать не более трех цветов: один для фона, один для заголовка, один для текста.</li><li>- Для фона и текста используйте контрастные цвета.</li><li>- Обратите внимание на цвет гиперссылок.</li></ul>   |
| Анимационные эффекты | <ul style="list-style-type: none"><li>- Используйте возможности компьютерной анимации для представления информации на слайде.</li><li>- Не стоит злоупотреблять различными анимационными эффектами, они не должны отвлекать внимание от содержания информации на слайде.</li></ul>  |

Представление информации:

|                                     |  |
|-------------------------------------|--|
| Содержание информации               | <ul style="list-style-type: none"><li>- Используйте короткие слова и предложения.</li><li>- Минимизируйте количество предлогов, наречий, прилагательных.</li><li>- Заголовки должны привлекать внимание аудитории.</li></ul>   |
| Расположение информации на странице | <ul style="list-style-type: none"><li>- Предпочтительно горизонтальное расположение информации.</li><li>- Наиболее важная информация должна располагаться в центре экрана.</li><li>- Если на слайде располагается картинка, надпись должна располагаться под ней.</li></ul>  |
| Шрифты                              | <ul style="list-style-type: none"><li>- Для заголовков – не менее 24.</li><li>- Для информации не менее 18.</li><li>- Шрифты без засечек легче читать с большого расстояния.</li><li>- Нельзя смешивать разные типы шрифтов в одной презентации.</li><li>- Для выделения информации следует использовать жирный шрифт, курсив или подчеркивание.</li><li>- Нельзя злоупотреблять прописными буквами.</li></ul> |

|                              |  |
|------------------------------|--|
| Способы выделения информации | Следует использовать:<br>-рамки; границы, заливку;<br>- штриховку, стрелки;<br>- рисунки, диаграммы, схемы для иллюстрации наиболее важных фактов.   |
| Объем информации             | -Не стоит заполнять один слайд слишком большим объемом информации: люди могут одновременно запомнить не более трех фактов, выводов, определений.<br>- Наибольшая эффективность достигается тогда, когда ключевые пункты отображаются по одному на каждом отдельном слайде. |
| Виды слайдов                 | Для обеспечения разнообразия следует использовать разные виды слайдов:<br>-с текстом;<br>- с таблицами, диаграммами.   |

#### Критерии оценки презентации

Общая оценка за презентацию выставляется следующим образом: если студент выполнил от 65% до 80% указанных выше требований, ему ставится минимальный балл – 3 (удовлетворительно); 80-90% — средний балл – 4 (хорошо); 90-100% — максимальный балл – 5 (отлично).

### 3. Методические рекомендации по написанию доклада

Доклад должен быть подготовлен на русском языке. Объем текста от 8 до 12 стр. (от 5000 до 30000 знаков без учета пробелов)

Для набора текста использовать программу Microsoft Word версии не выше 2003, для набора формул – Microsoft Equation 3.0.

В тексте статьи нужно использовать только международную систему единиц измерений (СИ).

Поля:

верхнее – 25мм;

нижнее – 20 мм;

левое – 30 мм;

правое – 20 мм;

красная строка 12,5 мм.

Бумага белая для офисной техники формата 210 x 297.

Качество печати – высокое.

В целях обеспечения унификации текстов в сборнике трудов конференции необходимо при оформлении доклада соблюдать следующие требования:

а) Название работы должно удовлетворять следующим требованиям: шрифт - Times New Roman, размер шрифта – 14, стиль – обычный, масштаб – 100%, интервал – обычный, смещение – нет, всё название должно быть написано жирным шрифтом и буквы должны быть прописными. Тип распределения текста на странице – по центру. Заголовки разделов оформляются аналогично названию работы, но только с одной заглавной, остальные буквы строчные.

б) После названия работы должны быть перечислены инициалы и фамилии авторов, а также с новой строки - название учреждения, которое данные авторы представляют. Требования к написанию: шрифт - Times New Roman, размер шрифта – 14, стиль – обычный, масштаб – 100%, интервал – обычный, смещение – нет, должно быть написано курсивом. Тип распределения текста на странице – по ширине.

в) Между названием и данными об авторе должна быть одна пустая строка.

г) Весь основной текст доклада набирается со следующими параметрами: шрифт - Times New Roman, размер шрифта – 14, стиль – обычный, масштаб – 100%, интервал – обычный, смещение – нет. Тип распределения текста на странице – по ширине.

д) Междустрочные интервалы принять одинарными.

е) При необходимости набора формул должен быть использован формульный редактор – Microsoft Equation 3.0:

шрифт Times New Roman или Symbol;

кегель -12;

надстрочные и подстрочные индексы кегль – 9.

положение по горизонтали должно совпадать с положением текста соответствующей строки

формулы, записанные в отдельной строке, должны располагаться по центру, а номер их – подогнан к правой границе листа.

ж) Надписи на рисунках и подрисуночные надписи, обозначения физических величин и их единиц, другие данные, помещаемые в таблицы, а также заголовки таблиц граф – шрифту 14.

з) Номера рисунков как в основном тексте так и в подрисуночной надписи приводить к виду – Рис. ....(№. Рисунка). Подрисуночная надпись должна располагаться вне “тела” рисунка.

и) Иллюстративный материал представляется только черно-белый.

Графические иллюстрации - в формате \*.wmf (векторная графика) или \*.tif (с глубиной цвета 1 бит), фотографии - в растровом формате \*.tif. При этом все элементы на иллюстрациях должны быть четкими, а надписи - свободно читаемыми; располагаться в центре листа.

к) Подрисуночная надпись аналогично располагается в центре листа. Номер таблицы указывается в виде – Таблица №, располагается по правой границе листа над самой таблицей, между таблицей и её номером должна быть одна пустая строка.

л) Список использованной литературы должен быть составлен по порядку ссылок в тексте. Необходимо представлять полный список, касающийся рассматриваемой темы. Настоятельно рекомендуется исключать самоцитирование.

Критерии оценки доклада

К *общим критериям* можно отнести:

соответствие доклада теме.

глубина и полнота раскрытия темы.

адекватность передачи первоисточника.

логичность, связность.

доказательность.

структурная упорядоченность (наличие введения, основной части, заключения, их оптимальное соотношение).

оформление (наличие оглавления, списка литературы, культура цитирования, сноски и т.д.).

языковая правильность.

Общая оценка за доклад выставляется следующим образом: если студент выполнил от 65% до 80% указанных выше требований, ему ставится минимальный балл – 3 (удовлетворительно); 80-90% — средний балл – 4 (хорошо); 90-100% — максимальный балл – 5 (отлично).

#### **4. Методические рекомендации по заполнению таблиц**

Таблица (из лат. *tabula* «доска») — способ передачи содержания, заключающийся в организации структуры данных, в которой отдельные элементы помещены в ячейки, каждой из которых сопоставлена пара значений — номер строки и номер колонки. Таким



образом, устанавливается смысловая связь между элементами, принадлежащими одному столбцу или одной строке.

Таблицы являются удобной формой для отображения информации. Но таблицы выполняют лишь тогда свою цель, когда между строчками и столбцами имеется смысловая связь, то есть информацию в них можно рассортировать неким образом, например, по дате или алфавиту.

Алгоритм заполнения таблицы.

Прочтите названия оглавлений таблицы.

Прочтите текст учебника и с помощью карандаша, укажите в нем материалы к каждой графе.

3. Запишите в соответствующие графы таблицы указанные материалы из текста в сокращенном виде.

Критерии оценки результата

| Уровни освоения                | Характеристика уровня                      |
|--------------------------------|--|
| Допустимый (удовлетворительно) | - Таблица заполнена верно на 50%           |
| Высокий (хорошо)               | - Таблица заполнена верно более чем на 50% |
| Оптимальный (отлично)          | - Таблица заполнена в полном объеме.       |

## 5. Методические рекомендации по составлению схем

Схемы как графические документы (графическая модель системы), на которых в виде условных обозначений или изображений показаны составные части некоторой системы и связи между ними.

Алгоритм составления схем

Прочтите предложенный текст и запишите его название

Укажите карандашом в книге основные разделы, из которых состоит текст и дайте им названия.

Проведите от названия текста стрелки вниз и подпишите возле каждой из них названия разделов текста.

Дополните схему примерами.

Критерии оценки результата

| Уровни освоения                | Характеристика уровня                                       |
|--------------------------------|---|
| Допустимый (удовлетворительно) | - все элементы присутствуют, отсутствует логика составления |
| Высокий (хорошо)               | - схема составлена с небольшими упущениями                  |

|                          |                                     |
|--------------------------|-------------------------------------|
| Оптимальный<br>(отлично) | - схема составлена логически верно. |
|--------------------------|-------------------------------------|

## 6. Методические рекомендации по составлению конспекта

Конспект - это последовательная фиксация информации, отобранной и обдуманной в процессе чтения.

Конспект:

подразумевает объединение плана, выписок и тезисов;

показывает внутреннюю логику изложения;

содержит основные выводы и положения, доказательства, приемы;

отражает отношение составителя к материалу;

может использоваться не только самим автором (составителем), но и другими читателями.

Основные требования к написанию конспекта: системность и логичность изложения материала, краткость, убедительность и доказательность.

При составлении конспекта необходимо избегать многословия, излишнего цитирования, стремления сохранить систематическую особенность текста в ущерб его логике.

Виды конспектов графически представлены на рис. 1.



Рис. 1. Виды конспектов

Общий алгоритм конспектирования состоит в следующем:

Общий алгоритм конспектирования состоит в следующем:

прочитать текст, отметить в нём новые слова, непонятные места, имена, даты; составить перечень основных мыслей, содержащихся в тексте, составить простой план, который поможет группировать материал в соответствии с логикой изложения;

выяснить в словаре значение новых непонятных слов, выписать их в тетрадь или словарь в конце тетради;

вторично прочитать текст, сочетая чтение с записью основных мыслей автора и их иллюстраций. Запись ведется своими словами, не переписывая текст. Важно стремиться к краткости, пользуясь правилами записи текста;

прочитать конспект ещё раз, доработать его.

Вместе с тем существуют некоторые особенности создания конспектов различных видов. Остановимся кратко на этом вопросе.

*Конспектирование* - процесс мысленной переработки и письменной фиксации информации, в виде краткого изложения основного содержания, смысла какого-либо текста.

*Выделение главной мысли* - одна из основ умственной культуры при работе с текстом. «Отбирать полезнейшее, - писал великий чешский педагог XVII века Я.А.Коменский, - дело такой важности, что немислим толковый читатель, без умения отбирать. Единственно надежный плод чтения - усвоение прочитанного, выбор полезного. Поистине только это держит ум в напряжении, запечатляет воспринятое в памяти и озаряет ум все более ярким светом. Не пожелать выделить из книги ничего, значит все пропустить».

*Результат конспектирования* - запись, позволяющая конспектирующему немедленно или через некоторый срок с нужной полнотой восстановить полученную информацию. Конспект в переводе с латыни означает «обзор». По существу его и составлять надо как обзор, содержащий основные мысли текста без подробностей и второстепенных деталей. Конспект носит индивидуализированный характер: он рассчитан на самого автора и поэтому может оказаться малопонятным для других.

План-конспект - это сжатый в форме плана пересказ прочитанного или услышанного.

Характеристика конспекта: краток, прост, быстро составляется и заполняется. Положительной чертой этого вида конспектов является то, что он учит выбирать главное, чётко и логично излагать мысли, даёт возможность усвоить материал ещё в процессе его изучения. Всё это делает его незаменимым при быстрой подготовке доклада, выступления. Однако работать с ним через некоторое время трудно, так как плохо восстанавливается в памяти содержание материала.

Этапы работы:

Составь план прочитанного текста или воспользуйся готовым.

Разъясни кратко и доказательно каждый пункт плана, выбери разумную и эффективную форму записи.

Сформулируй и запиши вывод.

План-конспект может выглядеть как таблица. Например:

| Основные вопросы  | Раскрытие вопросов  |
|---|---|
| 1. Сущность, содержание, основные характеристики бренда.                                  | Бренд — это атрибуты фирмы или товара, которые отражают их индивидуальность, привлекают внимание клиентов, создают имидж фирме, репутацию, способствуя продвижению товара на рынках.<br>Российскими авторами бренд трактуется как раскрученная торговая марка.<br>Символ бренда должен:<br>наиболее точно и полно отражать содержание товара;<br>обеспечить максимальное отличие от конкурентных брендов;<br>сформировать у потребителя убеждения, что этот бренд уникальный.<br>Уникальность символа бренда — главное требование при формировании бренда |
| 2. Взаимосвязь бренда с экономическими категориями «потребность», «спрос», «предложение». |   |
| 3. Технологии.  |   |
| 4. Управление активами бренда и др.   |   |

| Ключевые слова | Суть,<br>основная мысль | Раскрытие основной<br>мысли | Заключение,<br>вопросы, личные<br>отношения |
|----------------|-------------------------|-----------------------------|---|
|                |                         |                             |   |

Задание для работы при этом может быть сформулировано следующим образом:

Вариант 1. Внимательно прочтите предложенный текст (тексты) в учебнике (учебниках или распечатке). Представьте его в виде конспекта. На его основе составьте тезисы и план.

Вариант 2. Внимательно прочтите предложенный текст в учебнике или распечатке. Законспектируйте его, используя предложенный преподавателем план. Оформите план-конспект.

Вариант 3. Внимательно прочтите предложенный текст в учебнике или распечатке. Законспектируйте его, используя вид конспекта - тематический обзорный (раскрывает конкретную тему использованием нескольких источников).

Цитатный конспект — это конспект, созданный из цитат.

Характеристика конспекта: строится из высказываний автора, из изложенных им фактов. Чаще всего этот вид конспекта используется для работы с первоисточником. К нему студент может обращаться неоднократно. Но он не способствует актив мыслительной работе, поэтому, как правило, служит только иллюстрацией к изучаемой теме.

Этапы работы:

Прочитать текст, отметить в нём основное содержание, главные мысли, выделить те цитаты, которые войдут в конспект.

Пользуясь правилами сокращения цитат, выписать их в тетрадь. Форма записи может быть разной, например:

... (цитата);

... (цитата); (вывод);

основные вопросы; доказательства (цитаты); выводы.

Прочитать написанный текст, сверить его с оригиналом.

Сделать общий вывод.

Опорный конспект — это отражение изложения информации заложенной в тексте в виде опорных сигналов - слов, условных знаков, рисунков.

Характеристика конспекта: краток, учит выбирать главное, наглядно отражает причинно-следственные связи, развивает логическое мышление и образное умение моделировать информацию. Незаменим при повторении материала к зачёту, экзамену.

Этапы работы:

Прочитать внимательно текст.

Разделить его на смысловые части - блоки.

Поставить к каждой части вопрос.

Ответить на поставленный вопрос опорными сигналами, расположив их в виде логической схемы.

Свободный конспект — это сочетание выписок, цитат, тезисов.

Характеристика конспекта: он требует серьёзных усилий от студента при составлении, так как требует умений активного использования всех типов записей: планов, тезисов, выписок. Однако именно этот вид конспектов в высшей степени способствует прочному усвоению учебного материала.

Этапы работы:

Используя имеющиеся источники, выбрать материал по интересующей теме, изучить его и глубоко осмыслить.

Сделать необходимые выписки основных мыслей, цитат, составить тезисы.

Используя подготовленный материал, сформулировать основные положения по теме.

Тематический конспект — это конспект ответа на поставленный вопрос или конспект учебного материала по определенной теме.

Характеристика конспекта: он может быть обзорным и хро. но логическим; учит анализировать различные точки зрения на один и тот же вопрос, привлекать имеющиеся знания и личный опыт; используется в процессе работы над докладом, сообщением, рефератом.

Этапы работы:

Изучить несколько источников и сделать из них выборку материала по определённой теме или хронологии.

Мысленно оформить прочитанный материал в виде плана.

Пользуясь этим планом, кратко своими словами изложить осознанный материал.

Критерии результатов знаний и умений

«5» - уровень освоения студентом учебного материала достаточно высок, студент умеет использовать теоретические знания при выполнении практических задач с практикой, подтверждает сформированность общих и профессиональных компетенций;

«4» - студент полно освоил учебный материал, владеет понятийным аппаратом, ориентируется в изученном материале, осознанно применяет знания для решения практических задач, грамотно излагает ответ, но содержание и форма ответа имеют отдельные неточности;

«3» - студент знает и понимает основные положения учебного материала, но излагает его неполно, непоследовательно, допускает неточности в определении понятий, в применении знаний для решения практических задач не умеет доказательно обосновать свои суждения;

«2» - студент имеет разрозненные, бессистемные знания, не умеет выделять главное и второстепенное, допускает ошибки в определении понятий, искажает их смысл, беспорядочно и неуверенно излагает материал, не может применять знания для решения практических задач.

## **Критерии результатов знаний и умений**

### **1. Практические занятия.**

Оценка «отлично» ставится в том случае, если учащийся:

- выполнил работу в полном объеме с соблюдением необходимой
- последовательности проведения опытов и измерений;
- самостоятельно и рационально выбрал и подготовил для занятия необходимое оборудование, все работы провел в условиях и режимах, обеспечивающих получение результатов и выводов с наибольшей точностью;
- в представленном отчете правильно и аккуратно выполнил все записи, таблицы, рисунки, чертежи, графики, вычисления и сделал выводы;
- соблюдал требования безопасности труда.

Оценка «хорошо» ставится в том случае, если выполнены требования к оценке «5», но:

- было допущено два-три недочета, или не более одной негрубой ошибки и одного недочета.

Оценка «удовлетворительно» ставится, если работа выполнена не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы, или если в ходе проведения работы и измерений были допущены следующие ошибки:

- задание проводилось в нерациональных условиях, что привело к получению результатов с большим количеством ошибок,
- или в отчете были допущены в общей сложности не более двух ошибок (в записях единиц, измерениях, в вычислениях, графиках, таблицах, схемах, анализе

погрешностей и т. д.), не принципиального для данной работы характера, но повлиявших на результат выполнения,

- или не выполнен совсем или выполнен неверно анализ погрешностей;
- или работа выполнена не полностью, однако объем выполненной части таков, что позволяет получить правильные результаты и выводы по основным, принципиально важным задачам работы.

Оценка «неудовлетворительно» ставится в том случае, если:

- работа выполнена не полностью, и объем выполненной части работы не позволяет сделать правильных выводов,
- или в ходе работы и в отчете обнаружились в совокупности все недостатки, отмеченные в требованиях к оценке «3».

## **2. Практические занятия.**

Контрольная работа.

Оценка отлично ставится если обучающийся

- полно раскрыл содержание материала в объеме, предусмотренном рабочей программой,
- изложил материал грамотным языком в определенной логической последовательности, точно используя специальную терминологию;
- правильно выполнил рисунки, чертежи, сопутствующие ответу;
- отвечал самостоятельно без наводящих вопросов преподавателя. Возможны одна-две неточности при освещении второстепенных вопросов или в выкладках, которые студент легко исправил по замечанию преподавателя.

Оценка хорошо ставится если обучающийся

- в изложении допущены небольшие пробелы, не исказившие содержание ответа;
- допущены один – два недочета при освещении основного содержания ответа, исправленные по замечанию преподавателя;
- допущена ошибка или имеется более двух недочетов при освещении второстепенных вопросов или в выкладках, легко исправленные по замечанию преподавателя.

Оценка удовлетворительно ставится если обучающийся

- неполно или непоследовательно раскрыто содержание материала, но показано общее понимание вопроса;
- имелись затруднения или допущены ошибки в определении понятий, использовании терминологии и выкладках (определениях), исправленные после нескольких наводящих вопросов преподавателя;
- при знании теоретического материала выявлена недостаточная сформированность основных умений и навыков.

Оценка неудовлетворительно ставится если обучающийся

- не раскрыл основное содержание учебного материала;
- обнаружено незнание или непонимание студентом большей или наиболее важной части учебного материала;
- допущены ошибки в определении понятий, при использовании терминологии, в выкладках, которые не исправлены после нескольких наводящих вопросов преподавателя.

Тестовое задание

- Оценка отлично ставится если обучающийся ответил более чем на 85% вопросов.
- Оценка хорошо ставится если обучающийся ответил на 75-84% вопросов.
- Оценка хорошо ставится если обучающийся ответил на 74-60% вопросов.

- Оценка хорошо ставится если обучающийся ответил менее чем на 59% вопросов.