

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Волхонов Михаил Станиславович

Должность: Вице ректор

Дата подписания: 04.10.2023 17:22:34

Уникальный программный ключ:

b2dc75470204bc2bfce584f577a1b983ea237ea27559d45aa8272df0610c6a81

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«КОСТРОМСКАЯ ГОСУДАРСТВЕННАЯ СЕЛЬСКОХОЗЯЙСТВЕННАЯ АКАДЕМИЯ»

Утверждаю:  
декан электроэнергетического факультета

\_\_\_\_\_ /А.В. Рожнов/

14 июня 2023 года

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ  
по дисциплине  
**«Прикладное программирование»**

|                          |  |
|--------------------------|--|
| Направление подготовки   | <u>35.03.06 Агроинженерия</u>                        |
| Направленность (профиль) | <u>Информационные технологии в электроэнергетике</u> |
| Квалификация выпускника  | <u>бакалавр</u>                                      |
| Форма обучения           | <u>очная</u>   |
| Срок освоения ОПОП ВО    | <u>4 года</u>  |

Фонд оценочных средств предназначен для оценивания сформированности компетенций по дисциплине «Прикладное программирование».

Разработчик:

доцент кафедры информационных технологий в электроэнергетике

Яблоков А.С. \_\_\_\_\_

Утвержден на заседании кафедры информационных технологий в электроэнергетике, протокол №8 от «25» апреля 2023 года.

Заведующий кафедрой Солдатов В.А. \_\_\_\_\_

Согласовано:

Председатель методической комиссии электроэнергетического факультета протокол №5 от «13» июня 2023 года.

Яблоков А.С. \_\_\_\_\_

## Паспорт фонда оценочных средств

**Таблица 1**

| Модуль дисциплины  | Формируемые компетенции или их части  | Оценочные материалы и средства                  | Количество |
|--|---|---|------------|
| Введение в языки программирования, обзор. Элементы языка   | ПКос-2. Способен осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Тестирование                                    | 25         |
| Знакомство со средой программирования  |   | Тестирование                                    | 25         |
| Операторы языка  |   | Тестирование                                    | 25         |
| Понятие подпрограммы. Общие сведения о стандартных функциях. Разработка собственных функций пользователя                 |   | Индивидуальное домашнее задание<br>Тестирование | 25<br>30   |
| Структурированные типы данных. Массив, строка, структура, перечисление. Использование подпрограмм при обработке массивов |   | Тестирование                                    | 50         |
| Указатели и ссылки<br>Адресная арифметика, операции взятия адреса  |   | Тестирование                                    | 25         |
| Использование файлов при разработке информационных систем  |   | Индивидуальное домашнее задание<br>Тестирование | 10<br>10   |
| Объектно-ориентированное программирование  |   | Тестирование                                    | 25         |

# 1 ОЦЕНОЧНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ ДЕЯТЕЛЬНОСТИ В ПРОЦЕССЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

**Таблица 2 – Формируемые компетенции**

| Код и наименование компетенции  | Код и наименование индикатора достижения компетенции (части компетенции)  | Оценочные материалы и средства                  |
|---|---|---|
| 1   | 2   | 3   |
| <b>Введение в языки программирования, обзор. Элементы языка</b>   |   |   |
| ПКос-2. Способен осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | ИД-1 <sub>ПКос-2</sub> Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Тестирование                                    |
| <b>Знакомство со средой программирования</b>  |   |   |
| ПКос-2. Способен осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | ИД-1 <sub>ПКос-2</sub> Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Тестирование                                    |
| <b>Операторы языка</b>  |   |   |
| ПКос-2. Способен осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | ИД-1 <sub>ПКос-2</sub> Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Тестирование                                    |
| <b>Понятие подпрограммы.<br/>Общие сведения о стандартных функциях. Разработка собственных функций пользователя</b>                                   |   |   |
| ПКос-2. Способен осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | ИД-1 <sub>ПКос-2</sub> Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Индивидуальное домашнее задание<br>Тестирование |
| <b>Структурированные типы данных. Массив, строка, структура, перечисление.<br/>Использование подпрограмм при обработке массивов</b>                   |   |   |
| ПКос-2. Способен осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | ИД-1 <sub>ПКос-2</sub> Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Тестирование                                    |

| 1   | 2   | 3   |
|---|---|---|
| <b>Указатели и ссылки. Адресная арифметика, операции взятия адреса</b>  |   |   |
| ПКос-2. Способен осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | ИД-1 <sub>ПКос-2</sub> Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Тестирование                                    |
| <b>Использование файлов при разработке информационных систем</b>  |   |   |
| ПКос-2. Способен осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | ИД-1 <sub>ПКос-2</sub> Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Индивидуальное домашнее задание<br>Тестирование |
| <b>Объектно-ориентированное программирование</b>  |   |   |
| ПКос-2. Способен осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | ИД-1 <sub>ПКос-2</sub> Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Тестирование                                    |

### Оценочные материалы и средства для проверки сформированности компетенций

**Модуль: Введение в языки программирования, обзор. Элементы языка**

#### **Компьютерное тестирование (ТСк)**

*Выберите один правильный вариант ответа*

#### **Минимально значимая единица текста программы**

- +лексема
- пробел
- бит
- байт

#### **В каком из вариантов идентификатор построен правильно?**

- ЗvarN
- var\*NЗ
- +varNЗ
- переменнаяNЗ

#### **В идентификаторах языка C++ допускается**

- использовать кириллицу, цифры и символ подчеркивания
- +использовать латинские символы, цифры и символ подчеркивания

использовать пробелы, точки, символы \*, /, +, =  
начинать идентификатор с цифры

**Областью памяти компьютера, которой присвоено определенное имя, называют ...**

- +переменную
- ключевые слова
- знаки операций
- символьную константу

**#include в языке с++ - это**

- +директива компилятора, которая служит для подключения библиотек
- директива компилятора, которая служит для выбора условий при компиляции
- директива компилятора, которая используется для предотвращения повторного подключения библиотек
- оператор вывода на экран

**В результате выполнения оператора *cout* <<*a*;**

- выводится на экран символ 'a'
- +выводится на экран значение переменной "a"
- считывается переменная "a" с клавиатуры
- значение переменной "a" увеличивается на единицу

**Символьные константы в языке С++ ...**

- +записываются в одинарных кавычках
- записываются в двойных кавычках
- записываются без кавычек
- не существуют

**В языке С++ каждый оператор заканчивается знаком**

- ,
- ^
- ~
- +;

**Целочисленным типом данных языка С++ является**

- +int
- char
- double
- string

**К вещественному типу данных в языке С++ относится**

- int
- +double
- char

string

**К строковому типу данных языка C++ относится**

int

double

char

+string

**К логическому типу данных языка C++ относится**

+bool

double

char

string

**Выражение x++ осуществляет**

+увеличение переменной x на единицу

уменьшение переменной x на единицу

вывод переменной x на экран

считывание этой переменной с клавиатуры

**В каком из вариантов правильно объявлена символическая константа?**

const a = 7

const a[7]

const int a; a=7

+const int a=7

**Чему в результате выполнения представленной части программы будет равна переменная "y"?**

```
int a = 5;
```

```
y = 5 / 2;
```

2.5

+2

1

8

**Чему в результате выполнения представленной части программы будет равна переменная "y"?**

```
double a = 5;
```

```
y = 5 / 2;
```

+2.5

2

1

8

**bool a=true, b = false;**

**bool c=a&& b;**

**Чему будет равна переменная "c"?**

+false

true

may be

ни один из вариантов не подходит

**bool a=true, b = false;**

**bool c=a||b;**

**Чему будет равна переменная "c"?**

false

+true

may be

ни один из вариантов не подходит

**bool a=true, b = false;**

**bool c=!a;**

**Чему будет равна переменная "c"?**

+false

true

may be

ни один из вариантов не подходит

**Что возвращает функция *abs()*?**

квадрат числа

+модуль числа

корень числа

округленное число до ближайшего целого

**В библиотеке *cmath* находятся**

+математические функции

операторы ввода и вывода

функции для получения даты и времени

функции для генерации случайных чисел

**В каком из перечисленных вариантов представлена строковая константа?**

1984

19.84

+ "1984"

'4'

**В языке C++ символами // обозначается**

+комментарий



целочисленное деление  
 оператор вывода на экран  
 команда переключения языка

**Функция библиотеки *cmath* для нахождения квадратного корня — это...**

abs()  
 +sqrt()  
 pow()  
 koren()

**Найти остаток от деления целочисленных переменных можно с помощью операции...**

/  
 +%  
 \*  
 &

**Таблица 3 – Критерии оценки сформированности компетенций**

| Код и наименование индикатора достижения компетенции (части компетенции)   | Критерии оценивания сформированности компетенции (части компетенции)   |   |  |
|--|--|---|--|
|  | на базовом уровне  | на повышенном уровне  |  |
|  | соответствует оценке «удовлетворительно»<br>50-64% от максимального балла  | соответствует оценке «хорошо»<br>65-85% от максимального балла  | соответствует оценке «отлично»<br>86-100% от максимального балла   |
| ИД-1пКос-2<br>Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание основных правил, синтаксиса и типов данных языка программирования, на базовом уровне может осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание и понимание основных правил, синтаксиса и типов данных языка программирования, может самостоятельно осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует глубокое знание и понимание основных правил, синтаксиса и типов данных языка программирования, способен с высокой долей самостоятельности осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей |

## Модуль: Знакомство со средой программирования

### Компьютерное тестирование (ТСк)

Выберите один правильный вариант ответа

*int a=5, b=4;*

*int sum=a+b;*

Как должен выглядеть оператор вывода, чтобы ответ на экран выводился в следующем формате:  $a+b=9$ ?

```
cout<<a+b<<"="<<sum;  
+cout<<"a+b"<<"="<<sum;  
cout<<a<<"+"<<b<<"="<<sum;  
cout << sum>>
```

*int a=5, b=4;*

*int sum=a+b;*

Как должен выглядеть оператор вывода, чтобы ответ на экран выводился в следующем формате:  $5+4=9$ ?

```
cout<<a+b<<"="<<sum;  
cout<<"a+b"<<"="<<sum;  
+cout<<a<<"+"<<b<<"="<<sum;  
cout << sum>>
```

*int var=5;*

*cout<<"var";*

Что будет выведено на экран после выполнения этих строк кода?

+var

5

int

ничего

*int var=5;*

*cout<<var;*

Что будет выведено на экран после выполнения этих строк кода?

var

+5

int

ничего

Как, согласно стандарту языка C++, должен выглядеть заголовок главной функции *main*?

+int main()

void main()

int() main

int MAIN()

**Что возвращает функция *main* после успешного выполнения программы?**

1

+0

ничего не возвращает

долги

**Директива *#define* осуществляет..**

+макроподстановку

проверку условия

вывод переменной на экран

**Где должен записываться код программы?**

+внутри тела функции *main*

между ключевыми словами *begin* и *end*

внутри кавычек после директивы *include*

**К какому типу языков относится C++?**

интерпретируемые

+компилируемые

устной речи

**Интегрированная среда разработки включает:**

+компилятор, текстовый редактор, отладчик

текстовый редактор, графический редактор, редактор таблиц

компилятор, редактор таблиц, отладчик, загрузчик

**В результате сборки программы получается...**

объектный файл

+исполняемый файл

тестовый файл

**Как называется библиотека потоков консольного вывода?**

*cmath*

*fstream*

+*iostream*

*std*

**Что делает строка *using namespace std*?**

+сообщает компилятору, что дальше используется пространство имен *std*

сообщает компилятору, что это файл библиотеки

подключает библиотеку *std*

***int a, b;***

**Как осуществить чтение этих переменных из буфера клавиатуры?**

```
+cin>>a>>b  
cin<<a<<b  
cin<<a>>b  
cout<<a<<b
```

```
int a=6, b=7;
```

```
cout<<a<<b;
```

**Что будет на экране после выполнения этих строк?**

```
6 7  
a b  
+67  
ab
```

```
int a=6, b=7;
```

```
cout<<a<<" "<<b;
```

**Что будет на экране после выполнения этих строк?**

```
+6 7  
a b  
67  
ab
```

```
double a=2, b=8;
```

**Как вывести значение переменных a и b на экран, чтобы числа не сливались друг с другом?**

```
cout<<a<<b;  
cout<<a<<b<<endl;  
cout<<a <<b;  
cout<<a<<" "<<b;
```

**Для чего используется манипулятор *endl*?**

- +для перехода на новую строку
- для вывода чисел в десятичном формате
- для вывода чисел в шестнадцатеричном формате
- для задания ширины поля

**Для чего используется управляющая последовательность *\n*?**

- +для перехода на новую строку
- для вывода чисел в десятичном формате
- для вывода чисел в шестнадцатеричном формате
- для задания ширины поля

**Для чего используется манипулятор *hex*?**

- для перехода на новую строку

для вывода чисел в десятичном формате  
+для вывода чисел в шестнадцатеричном формате  
для задания ширины поля

### **Для чего используется манипулятор *setw*?**

для перехода на новую строку  
для вывода чисел в десятичном формате  
для вывода чисел в шестнадцатеричном формате  
+для задания ширины поля

### **Как ввести несколько чисел с клавиатуры (если программа ожидает их ввода)?**

числа разделяются пробелом  
после ввода каждого числа нужно нажимать клавишу *entr*  
+оба варианта правильные

**int a=5;**

### **Как вывести переменную *a* в шестнадцатеричном формате?**

```
cout<<a<<endl;  
+cout<<hex<<a<<endl;  
cout<<a<<hex<<endl;  
cin>>hex>>a;
```

**int a=8;**

### **Как вывести переменную *a* в восьмеричном формате?**

```
cout<<a<<endl;  
+cout<<oct<<a<<endl;  
cout<<a<<oct<<endl;  
cin>>oct>>a;
```

### **Какая библиотека используется для консольного потокового ввода и вывода?**

+iostream  
fstream  
cmath  
string

**Таблица 4 – Критерии оценки сформированности компетенций**

| Код и наименование индикатора достижения компетенции (части компетенции)   | Критерии оценивания сформированности компетенции (части компетенции)   |   |  |
|--|--|---|--|
|  | на базовом уровне  | на повышенном уровне  |  |
|  | соответствует оценке «удовлетворительно»<br>50-64% от максимального балла  | соответствует оценке «хорошо»<br>65-85% от максимального балла  | соответствует оценке «отлично»<br>86-100% от максимального балла   |
| ИД-1пКос-2<br>Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание основных приемов работы со средами разработки, назначения компиляторов, трансляторов и отладчиков, директив препроцессора, на базовом уровне может осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание и понимание основных приемов работы со средами разработки, назначения компиляторов, трансляторов и отладчиков, директив препроцессора, может самостоятельно осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует глубокое знание и понимание основных приемов работы со средами разработки, назначения компиляторов, трансляторов и отладчиков, директив препроцессора, способен с высокой долей самостоятельности осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей |

**Модуль: Операторы языка**

**Компьютерное тестирование (ТСк)**

*Выберите один правильный вариант ответа*

**Какая алгоритмическая конструкция задает выполнение того или иного оператора в зависимости от выполнения условия?**

- следование
- +ветвление
- цикл
- массив

**На какое максимальное количество ветвей позволяет разделить выполнение программы один оператор *if*?**

- одну
- +две
- четыре
- три

**Согласно синтаксическим правилам языка с++ логическое выражение (условие) оператора *if* пишется ...**

+ (в круглых скобках)

[в квадратных скобках]

{в фигурных скобках}

без скобок

**В языке С++ фигурные скобки используются для ...**

задания системы уравнений

для записи логического выражения условного оператора

+ объединения нескольких операторов в один блок

для комментариев

**Как в языке С++ записывается операция сравнения?**

+=

<>

=

<<

**Чему равна переменная *b* после выполнения описанного ниже кода программы?**

```
a=4;
```

```
if(a>3)
```

```
    b=4;
```

```
else
```

```
    b=1;
```

```
1
```

```
3
```

```
+4
```

```
5
```

**Чему равна переменная *b* после выполнения описанного ниже кода программы?**

```
b=3;
```

```
a=1;
```

```
if(a>3)
```

```
    b=4;
```

```
    b++;
```

```
1
```

```
3
```

```
+4
```

```
5
```

Чему равна переменная *b* после выполнения описанного ниже кода программы?

```
b=3;
a=1;
if(a>3)
{
  b=4;
  b++;
}
1
+3
4
5
```

Чему равна переменная *y* после выполнения описанного ниже кода программы?

```
a=2; b=3; x=1;
y=(x>2)?a+b:a-b;
-2
5
+-1
1
```

Какой оператор в языке C++ используется для разветвления вычислительного процесса на несколько направлений?

```
+switch
for
while
break
```

Чему будет равна переменная *b* после выполнения описанного ниже кода?

```
a=3;
switch (a)
{
  case 1: b=2; break;
  case 3: b=1; break;
  case 4: b=6; break;
  default: b=0;
}
2
+1
6
0
```



**Чему будет равна переменная  $b$  после выполнения описанного ниже кода?**

$a=7;$

***switch (a)***

***{***

***case 1: b=2; break;***

***case 3: b=1; break;***

***case 4: b=6; break;***

***default: b=0;***

***}***

2

1

6

+ 0

**Чему будет равна переменная  $b$  после выполнения описанного ниже кода?**

$a=1;$

***switch (a)***

***{***

***case 1: b=1;***

***case 3: b=2;***

***}***

+2

1

6

0

**Для организации многократно повторяющихся вычислений используют...**

следование

ветвление

+ циклы

**Когда в цикле с предусловием выполняется проверка логического выражения?**

+до выполнения тела цикла

после выполнения тела цикла

в теле цикла

никогда

**Когда в цикле с постусловием выполняется проверка логического выражения?**

до выполнения тела цикла

+после выполнения тела цикла

в теле цикла

никогда

**Один проход цикла называется ...**

+итерацией  
инкрементацией  
инкапсуляцией  
интерполяцией

**Сколько раз выполнится цикл?**

```
i=2;  
while(i<2)  
{  
    a+=i;  
    i++;  
}
```

+0

1

будет выполняться бесконечно

2

**Сколько раз выполнится цикл?**

```
i=2;  
do  
{  
    a+=i;  
    i++;  
} while(i<2);
```

0

+1

будет выполняться бесконечно

2

**Сколько раз выполнится цикл?**

```
i=1;  
do  
{  
    a+=i;  
} while(i<2);
```

0

1

+будет выполняться бесконечно

2

**Сколько раз выполнится цикл?**

```
for(int i=0; i<2;i++)
```

**$a+=i;$**   
0  
1  
будет выполняться бесконечно  
+2

**Выражение  $i<2$  в цикле  $for(int i=0; i<2;i++)$  является ...**

блоком инициализации  
+блоком проверки условия  
блоком модификации

**Если в теле цикла выполняется оператор  $break$ , то ...**

+происходит выход из тела цикла  
происходит переход к следующей итерации  
выполнение цикла становится на паузу до выполнения условия продолжения

**Если в теле цикла выполняется оператор  $continue$ , то ...**

происходит выход из тела цикла  
+происходит переход к следующей итерации  
выполнение цикла становится на паузу до выполнения условия продолжения

**Оператор безусловного перехода - ...**

+goto  
if  
break  
new

**Таблица 5 – Критерии оценки сформированности компетенций**

| Код и наименование индикатора достижения компетенции (части компетенции)   | Критерии оценивания сформированности компетенции (части компетенции)  |  |   |
|--|---|--|---|
|  | на базовом уровне   | на повышенном уровне   |   |
|  | соответствует оценке «удовлетворительно»<br>50-64% от максимального балла   | соответствует оценке «хорошо»<br>65-85% от максимального балла   | соответствует оценке «отлично»<br>86-100% от максимального балла  |
| ИД-1 <sub>ПКос-2</sub><br>Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание синтаксиса и работы операторов ветвления, цикла, управления, безусловного перехода, на базовом уровне может осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание и понимание синтаксиса и работы операторов ветвления, цикла, управления, безусловного перехода, может самостоятельно осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует глубокое знание и понимание синтаксиса и работы операторов ветвления, цикла, управления, безусловного перехода, способен с высокой долей самостоятельности осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей |

**Модуль: Понятие подпрограммы. Общие сведения о стандартных функциях. Разработка собственных функций пользователя**

**Компьютерное тестирование (ТСк)**

*Выберите один правильный вариант ответа*

**Разработка алгоритма решения задачи – это**

сведение задачи к математической модели, для которой известен метод решения  
выбор наилучшего метода из имеющихся

точное описание данных, условий задачи и ее целого решения

+определение последовательности действий, ведущих к получению результатов

**Алгоритм, в котором действия выполняются друг за другом, не повторяясь, называется:**

+линейным

циклическим

разветвленным

простым

**Именованная последовательность операторов, выполняющая какое-либо законченное действие, - это ...**

+функция  
переменная  
строка  
константа

**Можно ли в языке C++ объявлять одну функцию внутри другой?**

да  
+нет

**В каком из вариантов функция объявлена правильно?**

int add(int a, b)  
int 2add(int a, int b)  
+int add(int a, int b)  
int add{int a, int b}

**Результат какого типа возвращает данная функция?**

*int mf(double a)*  
+int  
double  
mf  
ни чего не возвращает

**Результат какого типа возвращает данная функция?**

*void mf(double a)*  
int  
double  
mf  
+ничего не возвращает

**Для чего в функции используется оператор *return*?**

+для выхода из функции и возврата значения  
только для выхода из функции  
чтобы передать аргумент по ссылке  
чтобы задать статическую переменную

**Какая из переменных данной функции является статической?**

```
int mf()  
{  
    int a;  
    static int b;  
    volatile int c;  
    int& e;  
}
```

a

+b  
c  
e

**Переменные без модификаторов, объявленные внутри тела функции, являются ...**

глобальными  
+локальными  
динамическими  
постоянными

**Как ограничивается время существования локальных переменных?**

+они создаются при вызове функции и уничтожаются при выходе из нее  
они создаются при первом вызове функции, но не уничтожаются при выходе из функции. Память освобождается при завершении программы  
они создаются при запуске программы до вызова функции и уничтожаются при выходе из нее  
они создаются при входе в функцию и больше ни когда не уничтожаются

**Как ограничивается время существования статических переменных?**

они создаются при вызове функции и уничтожаются при выходе из нее  
+они создаются при первом вызове функции, но не уничтожаются при выходе из функции. Память освобождается при завершении программы  
они создаются при запуске программы до вызова функции и уничтожаются при выходе из нее  
они создаются при входе в функцию и больше ни когда не уничтожаются

**По умолчанию в языке C++ аргументы передаются в функцию ...**

+по значению  
по ссылке  
только целочисленные  
только внутри массива

**Какой из аргументов данной функции передается по значению?**

***int myFunction(int a, int& b)***

```
{  
  int c=a+b;  
  char s;  
  return c;  
}
```

+a  
b  
c  
s

**Какой из аргументов данной функции передается по ссылке?**

```
int myFunction(int a, int& b)
```

```
{  
  int c=a+b;  
  char s;  
  return c;  
}  
a  
+b  
c  
s
```

**Какого оператора не может быть в функции, у которой тип возвращаемого значения void?**

```
+return  
if  
for  
break
```

**У какой функции нет прототипа?**

```
+main  
abs  
пользовательской  
встроенной
```

**В каком случае функция может изменить значение, переданного в нее фактического аргумента?**

```
при передаче аргумента по значению  
+при передаче аргумента по ссылке  
в любом случае сможет изменить  
в любом случае не изменит
```

**В каком случае НЕ функция может изменить значение, переданного в нее, фактического аргумента?**

```
+при передаче аргумента по значению  
при передаче аргумента по ссылке  
в любом случае сможет изменить  
в любом случае не изменит
```

**Для чего служит прототип функции?**

```
+сообщает компилятору о имени функции, возвращаемом значении, о типах и количестве аргументов, если вызов функции расположен перед ее определением
```

сообщает компилятору о имени функции, о количестве строк кода внутри функции, если вызов функции расположен перед ее определением используется для тестирования работы функции, а в последующем заменяется на конечный вариант

```
1 int add(int a, int b){
2     return a+b;
3 }
4 int main(){
5     int x=5, y=8;
6     int c;
7     c=add(x,y);
8     cout<<c;
9     return 0;
10 }
```

**В какой строке кода находится вызов функции *add*?**

1  
4  
+7  
9

```
1 int add(int a, int b){
2     return a+b;
3 }
4 int main(){
5     int x=5, y=8;
6     int c;
7     c=add(x,y);
8     cout<<c;
9     return 0;
10 }
```

**В какой строке кода находится определение функции *add*?**

+1  
4  
7  
9

**Можно ли в языке C++ вызывать одну функцию внутри другой?**

+да  
нет

**Как в языке C++ называется подпрограмма?**

+функция



массив  
структура  
указатель

**Как может выглядеть объявление функции, которая принимает в качестве аргумента строку и возвращает символ?**

+char fun(string s)  
int fun(string s)  
char fun(int s)  
string fun(char s)

```
int fun(int a, int b)
{
    a+b;
}
```

**Какие ошибки допущены при описании данной функции?**

попытка вернуть значение из функции с типом void  
тип указан не у всех аргументов  
имя функции не соответствует правилу составления идентификаторов  
+нет оператора возврата значения из функции

*Выберите несколько правильных вариантов ответа*

```
1 int add(int a, int b){
2     return a+b;
3 }
4 int main(){
5     int x=5, y=8;
6     int c;
7     c=add(x,y);
8     cout<<c;
9     return 0;
10 }
```

**Какие параметры функции add являются формальными?**

x  
y  
+a (50%)  
+b (50%)

```
1 int add(int a, int b){
2     return a+b;
3 }
4 int main(){
5     int x=5, y=8;
6     int c;
```

```
7   c=add(x,y);
8   cout<<c;
9   return 0;
10 }
```

**Какие параметры функции *add* являются фактическими?**

+x (50%)

+y (50%)

a

b

```
void fun(int a, b)
{
    return a+b;
}
```

**Какие ошибки допущены при описании данной функции?**

+попытка вернуть значение из функции с типом void (50%)

+тип указан не у всех аргументов (50%)

имя функции не соответствует правилу составления идентификаторов

нет оператора возврата значения из функции

```
void #fun(int a, int b)
{
    return a+b;
}
```

**Какие ошибки допущены при описании данной функции?**

+попытка вернуть значение из функции с типом void (50%)

тип указан не у всех аргументов

+имя функции не соответствует правилу составления идентификаторов (50%)

нет оператора возврата значения из функции

**Таблица 6 – Критерии оценки сформированности компетенций**

| Код и наименование индикатора достижения компетенции (части компетенции)   | Критерии оценивания сформированности компетенции (части компетенции)  |  |   |
|--|---|--|---|
|  | на базовом уровне   | на повышенном уровне   |   |
|  | соответствует оценке «удовлетворительно»<br>50-64% от максимального балла   | соответствует оценке «хорошо»<br>65-85% от максимального балла   | соответствует оценке «отлично»<br>86-100% от максимального балла  |
| ИД-1пКос-2<br>Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание принципов процедурного программирования, синтаксиса объявления и определения функций, на базовом уровне может осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание и понимание принципов процедурного программирования, синтаксиса объявления и определения функций, может самостоятельно осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует глубокое знание и понимание принципов процедурного программирования, синтаксиса объявления и определения функций, способен с высокой долей самостоятельности осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей |

**Модуль: Структурированные типы данных. Массив, строка, структура, перечисление. Использование подпрограмм при обработке массивов**

**Компьютерное тестирование (ТСк)**

*Выберите один правильный вариант ответа*

**В каком варианте переменная *a* не может быть строкой?**

```
char a[20];
string a;
char *a;
+char a;
```

**Какой символ выведется на экран после выполнения, приведенного ниже, кода?**

```
string a="Азбука";
cout << a.at(0);
+A
з
б
0
```

**Какая строка выведется на экран после выполнения, приведенного ниже, кода?**

```
string a = "Дом";  
a += "ик";  
cout << a;
```

Дом  
ик  
+Домик  
Дом ик

**Какой будет результат выполнения приведенного ниже кода?**

```
string a1 = "Дом";  
string a2 = "Лом";  
cout << (a1 < a2);
```

+true  
false  
Дом  
Лом

**Какой символ выведется на экран после выполнения приведенного ниже кода?**

```
string a = "Азбука";  
cout << a.at(a.length()-1);
```

+a  
А  
з  
0

**Как узнать длину строки типа *string*?**

с помощью функции *strlen*  
+с помощью метода *length*  
с помощью метода *find*  
с помощью метода *erase*

**Как осуществить поиск подстроки в строке типа *string*?**

с помощью функции *strlen*  
с помощью метода *length*  
+с помощью метода *find*  
с помощью метода *erase*

**Как удалить символы в строке типа *string*?**

с помощью функции *strlen*  
с помощью метода *length*  
с помощью метода *find*  
+с помощью метода *erase*

**Какую функцию выполняет операция «+» при работе с объектами string?**

сложение чисел

+конкатенацию строк

увеличения размера строки

**Можно ли в языке C++ сравнивать строки типа string с помощью операций сравнения(<,>==...)?**

+да

нет

**Можно ли в языке C++ сравнивать строки типа char\* с помощью операций сравнения(<,>==...)?**

да

+нет

**Какую функцию выполняет метод *at(i)* при работе со строками класса string?**

выполняет конкатенацию строк

+возвращает *i*-ый символ из строк

удаляет *i* символов в строке

осуществляет поиск подстроки *i* в строке

**Чему равно значение константы ERR\_\_WRITE?**

```
enum Err {ERR__READ, ERR__WRITE, ERR_CONVERT};
```

0

+1

2

3

**Чему равно значение константы ERR\_CONVERT?**

```
enum Err {ERR__READ, ERR__WRITE=3, ERR_CONVERT};
```

0

1

2

+4

**Могут ли совпадать имена констант внутри перечисления?**

да

+нет

**Могут ли совпадать значения констант внутри перечисления?**

+да

нет

**Как при объявлении перечисления (enum) записывается список констант?**

+в фигурных скобках через запятую  
в круглых скобках через запятую  
в круглых скобках через точку  
в квадратных скобках

**Сколько полей у структуры "student"?**

```
struct student
```

```
{
```

```
    string fio;
```

```
    int group;
```

```
    int mark;
```

```
};
```

1

2

+3

4

**Где правильно осуществляется доступ к полям структуры?**

```
struct student
```

```
{
```

```
    string fio;
```

```
    int group;
```

```
    int mark;
```

```
};
```

```
student a;
```

```
+cin >>a.fio;
```

```
fio(a)="ИВАНОВ";
```

```
cin>>a->fio;
```

```
cin>>a[1].fio;
```

**Где правильно осуществляется доступ к полям структуры?**

```
struct student
```

```
{
```

```
    string fio;
```

```
    int group;
```

```
    int mark;
```

```
};
```

```
student b; student *a=&b;
```

```
cin >>a.fio;
```

```
fio(a)="ИВАНОВ";
```

```
+cin>>a->fio;
```

```
cin>>a[1].fio;
```

**Где правильно осуществляется доступ к полям структуры?**

```
struct student  
{  
    string fio;  
    int group;  
    int mark;  
};  
student a[10];  
cin >>a.fio;  
fio(a)="ИВАНОВ";  
cin>>a->fio;  
+cin>>a[1].fio;
```

**Какое ключевое слово используется при описании структуры?**

+struct  
class  
string  
name

**Может ли поле структуры иметь тип этой же структуры?**

да  
+нет

**Может ли поле структуры быть указателем на эту же структуру?**

+да  
нет

```
struct  
{  
    int a;  
    int b;  
} test1, test2;
```

**Сколько всего существует объектов данной структуры?**

1  
+2  
3  
0

```
struct test  
{  
    int a;  
    int b;  
} test1, test2;
```

**Какого типа объекты test1 и test2?**

+test  
string  
int  
bool

**Если при определении структуры не создаются объекты этой структуры, то после фигурной скобки следует ставить...**

точку « . »  
+точку с запятой « ; »  
двоеточие « : »  
писать ключевое слов *end*

**Какой режим доступа устанавливается для полей структуры по умолчанию?**

доступ к полям запрещен  
+доступ к полям разрешен  
доступ разрешен только к полям структур, являющихся частью другой структуры

**Что такое массив?**

+совокупность переменных, имеющих один тип и объединенных под одним именем  
совокупность переменных разного типа, объединенных под одним именем  
совокупность констант, имеющих один тип и объединенных под одним именем

**Размер массива указывается в ...**

круглых скобках  
фигурных скобках  
+квадратных скобках

**Выберите правильное объявление массива**

int mas{5};  
mas int(5);  
mas int[5];  
+int mas[5];

**Нумерация массива начинается ...**

+с нуля  
с единицы  
может задаваться произвольно

**int m[10]; Какой индекс последнего элемента массива m?**

10  
+9  
8  
0



**int m[10];** Сколько элементов в массиве m?

+10

9

8

0

**При одновременных инициализации и объявлении массива его элементы записываются ...**

в круглых скобках

в квадратных скобках

+в фигурных скобках

**int m[5]={0, 6, 4, 7, 4};**

**cin<<m[2];**

**После выполнения данных строчек кода, на экран выведется число ...**

2

+4

6

ни одно из перечисленных, в данном коде происходит выход за границу массива

**int m[5]={0, 6, 4, 7, 4};**

**cin<<m[5];**

**После выполнения данных строчек кода, на экран выведется число ...**

2

4

6

+ни одно из перечисленных, в данном коде происходит выход за границу массива

**В каком из вариантов записано правильное заполнение массива?**

**double m[5]; for(int i=0; i<=5; i++) cin>>m[i];**

**+double m[5]; for(int i=0; i<5; i++) cin>>m[i];**

**double m[5]; for(int i=1; i<=5; i++) cin>>m[i];**

**double m[5]; for(int i=1; i<5; i++) cin>>m[i];**

**int m[10];**

**for(int i=0;i<10;i++) m[i]=rand();**

**for(int i=0;i<10;i++) cout<<m[i]<<endl;**

**При выполнении данных строчек кода массив выводится на экран ...**

в строку

+в столбец

по диагонали

```
int m[10];  
for(int i=0;i<10;i++) m[i]=rand();  
for(int i=0;i<10;i++) cout<<m[i];
```

**При выполнении данных строчек кода массив выводится на экран ...**

+в строку  
в столбец  
по диагонали

```
int m[10];  
for(int i=0;i<10;i++) m[i]=rand();  
int a;  
for(int i=0;i<10;i++) a=m[i];
```

**При выполнении данных строчек кода в переменной *a* будет сохранено...**

+значение последнего элемента массива  
значение максимального элемента массива  
индекс максимального элемента массива  
сумма всех элементов массива  
произведение всех элементов массива

```
int m[10];  
for(int i=0;i<10;i++) m[i]=rand();  
int a;  
for(int i=0;i<10;i++) if(a<m[i]) a=m[i];
```

**При выполнении данных строчек кода в переменной *a* будет сохранено...**

значение последнего элемента массива  
+значение максимального элемента массива  
индекс максимального элемента массива  
сумма всех элементов массива  
произведение всех элементов массива

```
int m[10];  
for(int i=0;i<10;i++) m[i]=rand();  
int a;  
for(int i=0;i<10;i++) if(a<m[i]) a=i;
```

**При выполнении данных строчек кода в переменной *a* будет сохранено...**

значение последнего элемента массива  
значение максимального элемента массива  
+индекс максимального элемента массива  
сумма всех элементов массива  
произведение всех элементов массива

```
int m[10];
for(int i=0;i<10;i++) m[i]=rand();
int a;
for(int i=0;i<10;i++) a+=m[i];
```

**При выполнении данных строчек кода в переменной *a* будет сохранено...**

значение последнего элемента массива  
значение максимального элемента массива  
индекс максимального элемента массива  
+сумма всех элементов массива  
произведение всех элементов массива

```
int m[10];
for(int i=0;i<10;i++) m[i]=rand();
int a;
for(int i=0;i<10;i++) a*=m[i];
```

**При выполнении данных строчек кода в переменной *a* будет сохранено...**

значение последнего элемента массива  
значение максимального элемента массива  
индекс максимального элемента массива  
сумма всех элементов массива  
+произведение всех элементов массива

**Выберите правильное объявление двумерного массива**

```
int m[3, 4];
+int m[3][4];
int m(3, 4);
int m(3)(4);
```

**Выберите правильное объявление двумерного массива с одновременной инициализацией**

```
int m[3][2]={{1, 2}, [3, 4], [5,6]};
+int m[3][2]={{1, 2}, {3, 4}, {5,6}};
int m[3, 2]={{1, 2}, [3, 4], [5,6]};
```

*Выберите несколько правильных вариантов ответа*

**Какие виды строк существуют в языке C++?**

+массив символов (50%)  
+объект класса string (50%)  
функция string  
перечисление enum string

**Как прочитать с клавиатуры строку, которая содержит несколько слов, разделенных пробелами?**

с помощью обычной операции `cin >>`

+с помощью метода `cin.get` (50%)  
 +с помощью функции `getline` (50%)  
 с помощью оператора `input`

**Какие операции можно использовать со строками типа `string`?**

+«+» (33%)  
 «-»  
 +«>» (33%)  
 +«==» (33%)

**Таблица 7 – Критерии оценки сформированности компетенций**

| Код и наименование индикатора достижения компетенции (части компетенции)   | Критерии оценивания сформированности компетенции (части компетенции)   |   |  |
|--|--|---|--|
|  | на базовом уровне  | на повышенном уровне  |  |
|  | соответствует оценке «удовлетворительно»<br>50-64% от максимального балла  | соответствует оценке «хорошо»<br>65-85% от максимального балла  | соответствует оценке «отлично»<br>86-100% от максимального балла   |
| ИД-1пКос-2<br>Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание синтаксиса и метода работы со строками, массивами, структурами и перечислениями, на базовом уровне может осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание и понимание синтаксиса и метода работы со строками, массивами, структурами и перечислениями, может самостоятельно осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует глубокое знание и понимание синтаксиса и метода работы со строками, массивами, структурами и перечислениями, способен с высокой долей самостоятельности осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей |

**Модуль: Указатели и ссылки Адресная арифметика, операции взятия адреса**

**Компьютерное тестирование (ТСк)**

*Выберите один правильный вариант ответа*

**Переменная, в которой хранится адрес другого объекта:**

+указатель  
 массив  
 показатель  
 адресат

**Выберите правильное объявление указателя**

int a;  
+int \*a;  
int &a;  
int \$a;

**Какой символ используется для записи операции взятия адреса?**

\*  
\$  
#  
+&

**Какой символ используется для записи операции разыменования указателя?**

+\*  
\$  
#  
&

**int a=5;  
int \*b=&a;**

**cout<<b;**

**Что будет выведено на экран?**

число 5  
+адрес переменной a  
адрес переменной b

**int a=5;  
int \*b=&a;**

**cout<<\*b;**

**Что будет выведено на экран?**

+число 5  
адрес переменной a  
адрес переменной b

**int a=5;  
int \*b=&a;  
\*b = \*b + 2;**

**Чему равна переменная a после выполнения этих строчек кода?**

5  
+7  
2  
0

```
int a=5;
int *b=&a;
b = b + 2;
```

**Чему равна переменная *a* после выполнения этих строчек кода?**

+5  
7  
2  
0

**Какие арифметические операции можно применять к указателю?**

деление и сложение  
умножение и вычитание  
+сложение и вычитание  
умножение и деление

**Что произойдет после применения к указателю операции инкремента?**

увеличится на единицу значение переменной, на которую он ссылается  
+хранящийся в указателе адрес увеличится на количество ячеек памяти базового типа  
операцию инкремента нельзя применять к указателю  
значение переменной, на которую он ссылается, увеличится на количество ячеек памяти базового типа

```
int m[10];
int *p;
```

**Как сделать так, чтобы указатель ссылался на массив?**

```
p=&m;
+p=m;
p=*m;
```

```
int m[]={1,2,3};
int *p=m;
cout<<*(p++);
```

**Что выведется на экран?**

число 1  
+число 2  
адрес нулевого элемента массива  
адрес первого элемента массива

**Выберите правильное объявление массива указателей:**

```
+int *m[10];
int m[*10];
int *m=[10];
```

```
void fun(int a, double b, char c, double *d);
```

Какой из аргументов данной функции является указателем?

- a
- b
- c
- +d

```
int add(int* a, int* b);
```

```
int main()
```

```
{
```

```
int x=5, y = 6, z;
```

Как будет выглядеть вызов функции?

```
z=add(x,y);
```

```
+z=add(&x,&y);
```

```
z=add(*x,*y);
```

```
z=*add(x,y);
```

Какой вид указателей используют для того, чтобы предотвратить возможность изменения значения переменной, на которую он ссылается?

- +указатель на константу
- константный указатель
- указатель на массив
- массив указателей

Какой вид указателей используют для того, чтобы было возможно изменить значения переменной, на которую он ссылается, но нельзя было изменить сам указатель?

- указатель на константу
- +константный указатель
- указатель на массив
- массив указателей

В какой строке правильно объявлен указатель на константу?

- +const int \*p=&a;
- int\* const p=&a;
- int \*p=const& a;

В какой строке правильно объявлен константный указатель?

- const int \*p=&a;
- +int\* const p=&a;
- int \*p=const& a;

Выберите правильное объявление ссылки:

- int a=b;

```
int *a=b;  
+int &a=b;  
int $a=b;
```

**По своей сути, ссылка является ...**

указателем на константу  
+константным указателем  
указателем на массив  
массивом указателей

**Какой символ используется для объявления ссылки?**

\*  
\$  
#  
+&

**Нужно ли использовать операцию разыменования при работе со ссылками?**

да  
+нет

```
int a=5, b=6;  
int& r=a;  
r +=2;  
cout<<r;
```

**Какое число будет выведено на экран?**

2  
5  
6  
+7

```
int a=5, b=6;  
int& r=a;  
r +=2;  
cout<<r;
```

**Чему станет равна переменная *a*?**

2  
5  
6  
+7



**Таблица 8 – Критерии оценки сформированности компетенций**

| Код и наименование индикатора достижения компетенции (части компетенции)   | Критерии оценивания сформированности компетенции (части компетенции)   |   |  |
|--|--|---|--|
|  | на базовом уровне  | на повышенном уровне  |  |
|  | соответствует оценке «удовлетворительно»<br>50-64% от максимального балла  | соответствует оценке «хорошо»<br>65-85% от максимального балла  | соответствует оценке «отлично»<br>86-100% от максимального балла   |
| ИД-1 <sub>ПКос-2</sub><br>Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание метода работы с указателями и ссылками, принципа действия адресной арифметики, на базовом уровне может осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание и понимание метода работы с указателями и ссылками, принципа действия адресной арифметики, может самостоятельно осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует глубокое знание и понимание метода работы с указателями и ссылками, принципа действия адресной арифметики, способен с высокой долей самостоятельности осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей |

**Модуль: Использование файлов при разработке информационных систем**

**Компьютерное тестирование (ТСк)**

*Выберите один правильный вариант ответа*

**Какая библиотека используется для работы с файловыми потоками ввода-вывода?**

- iostream
- +fstream
- cmath
- io manip

**Для генерации потока файлового ввода необходимо создать объект класса ...**

- +ifstream
- ofstream
- sstream
- iostream

**Для генерации потока файлового вывода необходимо создать объект класса:**

- ifstream

+ofstream  
sstream  
iostream

**В каком режиме открываются файлы по умолчанию?**

+текстовом  
бинарном  
мультимедийном

**С помощью какого оператора можно осуществить чтение из файлового потока?**

+>>  
<<  
^  
=

**С помощью какого оператора можно осуществить запись в файловый поток?**

>>  
+<<  
^  
=

**С помощью какой функции можно закрыть файл?**

+close  
open  
exit  
break

**С помощью какого флага можно открыть файл в бинарном режиме?**

+ios::binary  
ios::in  
ios::app  
ios::out

**С помощью какого флага задается режим ввода, при котором данные дописываются в конец файла?**

ios::binary  
ios::in  
+ios::app  
ios::out

**Какая функция позволяет переместиться в любое место файлового потока вывода?**

+seekp  
seekg  
move  
get

**Таблица 9 – Критерии оценки сформированности компетенций**

| Код и наименование индикатора достижения компетенции (части компетенции)   | Критерии оценивания сформированности компетенции (части компетенции)   |   |  |
|--|--|---|--|
|  | на базовом уровне  | на повышенном уровне  |  |
|  | соответствует оценке «удовлетворительно» 50-64% от максимального балла   | соответствует оценке «хорошо» 65-85% от максимального балла   | соответствует оценке «отлично» 86-100% от максимального балла  |
| ИД-1пКос-2<br>Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание основного подхода к использованию файлов в создаваемых программах, способы осуществления параллельного и последовательного доступа к файлам, на базовом уровне может осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание и понимание основного подхода к использованию файлов в создаваемых программах, способы осуществления параллельного и последовательного доступа к файлам, может самостоятельно осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует глубокое знание и понимание основного подхода к использованию файлов в создаваемых программах, способы осуществления параллельного и последовательного доступа к файлам, способен с высокой долей самостоятельности осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей |

**Модуль: Объектно-ориентированное программирование**

**Компьютерное тестирование (ТСк)**

*Выберите один правильный вариант ответа*

**Механизм, связывающий воедино код и данные, которыми он манипулирует, а также обеспечивающий их защиту от внешнего вмешательства и неправильного использования, называется...**

- +инкапсуляция
- полиморфизм
- наследование

**Атрибут, позволяющий с помощью одного интерфейса управлять доступом к целому ряду методов, называется ...**

инкапсуляция  
+полиморфизм  
наследование

**Механизм, который позволяет строить иерархии, в которых производные классы получают элементы родительских классов и могут дополнять их или изменять их свойства, называется...**

инкапсуляция  
полиморфизм  
+наследование

**Какое ключевое слово используется при описании класса в языке C++?**

struct  
+class  
union  
enum

**Какое утверждение является ошибочным?**

поля класса могут быть любого типа, кроме типа этого же класса  
поля класса могут быть описаны с модификатором const  
поля класса могут быть описаны с модификатором static  
+возможна инициализация полей класса при его описании

**Какой режим доступа у полей класса по умолчанию?**

public  
+private  
protected

**Какой элемент класса предназначен для инициализации объекта и вызывается автоматически при его создании?**

+конструктор  
деструктор  
дружественная функция  
указатель this

**Какой элемент класса предназначен для освобождения памяти, занимаемой объектом?**

конструктор  
+деструктор  
дружественная функция  
указатель this

**Как называется конструктор, который можно вызвать без параметров?**

+конструктор по умолчанию

копирующий конструктор  
деструктор  
закрытый конструктор

**Как называется конструктор, который вызывается при инициализации объекта другим объектом этого же класса?**

конструктор по умолчанию  
+копирующий конструктор  
деструктор  
закрытый конструктор

```
1 class Mclass
2 {
3     private:
4         int a;
5     public:
6         Mclass();
7         ~Mclass();
8         int b;
9         int get();
10};
```

**В какой строке объявлено закрытое поле класса?**

+4  
6  
7  
8  
9

```
1 class Mclass
2 {
3     private:
4         int a;
5     public:
6         Mclass();
7         ~Mclass();
8         int b;
9         int get();
10};
```

**В какой строке объявлено открытое поле класса?**

4  
6  
7  
+8  
9

```
1 class Mclass
2 {
3     private:
4         int a;
5     public:
6         Mclass();
7         ~Mclass();
8         int b;
9         int get();
10};
```

**В какой строке объявлен метод класса?**

4  
6  
7  
8  
+9

```
1 class Mclass
2 {
3     private:
4         int a;
5     public:
6         Mclass();
7         ~Mclass();
8         int b;
9         int get();
10};
```

**В какой строке объявлен конструктор класса?**

4  
+6  
7  
8  
9

**Как обычной функции (не методу класса) предоставить доступ к закрытым полям класса?**

объявить ее статической

+объявить ее дружественной

объявит ее константной

любая функция и так имеет доступ к закрытым полям класса

**С помощью какого ключевого слова можно заставить компилятор создать только один экземпляр поля класса, который будет использоваться всеми объектами?**

const  
+static  
virtual  
friend

**Где правильно осуществляется доступ к методам класса?**

```
class student
{
    string fio;
    int group;
    public:
        string getFio();
        void setGroup(int a);
};
student a;
+cout<<a.getFio();
cout<<getFio(a);
cout<<a->getFio();
cout<<a[1].getFio();
```

**Где правильно осуществляется доступ к методам класса?**

```
class student
{
    string fio;
    int group;
    public:
        string getFio();
        void setGroup(int a);
};
student *a;
cout<<a.getFio();
cout<<getFio(a);
+cout<<a->getFio();
cout<<a[1].getFio();
```

**Что в языке C++ означает ключевое слово *this*?**

+указатель на текущий объект класса  
указатель на метод класса  
спецификатор доступа

**С помощью какого ключевого слова объявляются виртуальные методы класса?**

const  
static  
+virtual  
friend

**При повышающем приведении типов, чтобы вместо метода базового класса вызывался метод производного класса, используют ...**

дружественные методы  
+виртуальные методы  
статические методы

**Выберите правильное объявление чисто виртуального метода:**

clear virtual f(int);  
+virtual f(int)=0;  
f(int) = virtual;

**Если внутри класса есть виртуальный метод, то класс становится ...**

+абстрактным  
константным  
статическим  
дружественным

*Выберите несколько правильных вариантов ответа*

**Какие ограничения накладываются на константные методы класса?**

они могут вызываться только для константных объектов  
+они могут вызывать только константные методы (50%)  
+они не могут менять значение полей класса (50%)  
они могут вызываться только внутри конструктора класса

**Какие требования выдвигаются при перегрузке функции?**

+перегруженные функции должны отличаться типом параметров (50%)  
+перегруженные функции должны отличаться количеством параметров (50%)  
перегруженные функции должны отличаться типом возвращаемого значения

**Какие элементы класса не наследуются?**

поля  
методы  
+конструкторы (33%)  
+деструкторы (33%)  
+оператор присвоения (33%)



**Выберите верные утверждения:**

виртуальные метод может объявляться с модификатором static

+виртуальные метод не может объявляться с модификатором static (50%)

+виртуальные методы наследуются (50%)

деструктор не может быть виртуальным

**Таблица 10 – Критерии оценки сформированности компетенций**

| Код и наименование индикатора достижения компетенции (части компетенции)   | Критерии оценивания сформированности компетенции (части компетенции)   |   |  |
|--|--|---|--|
|  | на базовом уровне  | на повышенном уровне  |  |
|  | соответствует оценке «удовлетворительно»<br>50-64% от максимального балла  | соответствует оценке «хорошо»<br>65-85% от максимального балла  | соответствует оценке «отлично»<br>86-100% от максимального балла   |
| ИД-1ПКос-2<br>Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание основ объектно-ориентированного программирования, синтаксиса определения классов, приемов использования наследования и полиморфизма, на базовом уровне может осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание и понимание основ объектно-ориентированного программирования, синтаксиса определения классов, приемов использования наследования и полиморфизма, может самостоятельно осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует глубокое знание и понимание основ объектно-ориентированного программирования, синтаксиса определения классов, приемов использования наследования и полиморфизма, способен с высокой долей самостоятельности осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей |

## 2 ОЦЕНИВАНИЕ ПИСЬМЕННЫХ РАБОТ СТУДЕНТОВ

### Оценивание письменных работ студентов, не регламентируемых учебным планом

**Модуль: Понятие подпрограммы. Общие сведения о стандартных функциях.  
Разработка собственных функций пользователя**

#### **Индивидуальное домашнее задание (ИДЗ)**

##### *Варианты задания*

- 1) Создать функцию для нахождения корней квадратного уравнения. Написать программу для тестирования этой функции.
- 2) Создать функцию для нахождения простых чисел в заданном диапазоне. Написать программу для тестирования этой функции.
- 3) Создать функцию для определения дальности полета снаряда, выпущенного под углом к горизонту. Написать программу для тестирования этой функции.
- 4) Создать функцию для нахождения чисел последовательности Фибоначчи в заданном диапазоне. Написать программу для тестирования этой функции.
- 5) Создать функцию для расчета частоты колебательного контура. Написать программу для тестирования этой функции.
- 6) Создать функцию для расчета времени переходного процесса в RC-цепи. Написать программу для тестирования этой функции.
- 7) Создать функцию для расчета времени переходного процесса в LC-цепи. Написать программу для тестирования этой функции.
- 8) Создать функцию для вывода случайной последовательности символов из заданного диапазона. Написать программу для тестирования этой функции.
- 9) Создать функцию для расчета потенциальной энергии заряженных предметов. Написать программу для тестирования этой функции.
- 10) Создать функцию для нахождения факториала числа. Написать программу для тестирования этой функции.
- 11) Создать функцию для расчета коэффициента мощности. Написать программу для тестирования этой функции.
- 12) Создать функцию для расчета кинетической энергии тела, брошенного вертикально вверх. Написать программу для тестирования этой функции.
- 13) Создать функцию для расчета сопротивления токового шунта. Написать программу для тестирования этой функции.
- 14) Создать функцию для расчета делителя напряжений. Написать программу для тестирования этой функции.
- 15) Создать функцию для перевода температуры из градусов Цельсия в градусы Фаренгейта. Написать программу для тестирования этой функции.
- 16) Создать функцию для перевода Джоулей в кВт\*ч. Написать программу для тестирования этой функции.
- 17) Создать функцию для нахождения наименьшего общего кратного. Написать программу для тестирования этой функции.

- 18) Создать функцию для нахождения наибольшего общего делителя. Написать программу для тестирования этой функции.
- 19) Создать функцию для расчета тока в шунте с заданным сопротивлением. Написать программу для тестирования этой функции.
- 20) Создать функцию для нахождения напряжения на резистивном делителе с заданным сопротивлением. Написать программу для тестирования этой функции.
- 21) Создать функцию для нахождения пифагоровых троек в заданном диапазоне. Написать программу для тестирования этой функции.
- 22) Создать функцию для нахождения объема призмы. Написать программу для тестирования этой функции.
- 23) Создать функцию для нахождения объема усеченного конуса. Написать программу для тестирования этой функции.
- 24) Создать функцию для расчета емкости плоского конденсатора. Написать программу для тестирования этой функции.
- 25) Создать функцию для расчета радиуса описанной окружности правильного многоугольника. Написать программу для тестирования этой функции.

Максимальная оценка – 10 баллов

**Таблица 11 – Критерии оценки сформированности компетенций**

| Код и наименование индикатора достижения компетенции (части компетенции)   | Критерии оценивания сформированности компетенции (части компетенции)   |  |   |
|--|--|--|---|
|  | на базовом уровне  | на повышенном уровне   |   |
|  | соответствует оценке «удовлетворительно»<br>50-64% от максимального балла  | соответствует оценке «хорошо»<br>65-85% от максимального балла   | соответствует оценке «отлично»<br>86-100% от максимального балла  |
| ИД-1 <sub>ПКос-2</sub><br>Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание принципов процедурного программирования, синтаксиса объявления и определения функций, может использовать процедурный подход и создавать собственные функции, на базовом уровне может осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание и понимание принципов процедурного программирования, синтаксиса объявления и определения функций, способен использовать процедурный подход и создавать собственные функции, самостоятельно осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует глубокое знание и понимание принципов процедурного программирования, синтаксиса объявления и определения функций, с высоким уровнем самостоятельности способен использовать процедурный подход и создавать собственные функции, способен с высокой долей самостоятельности осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей |

## **Модуль: Использование файлов при разработке информационных систем**

### **Индивидуальное домашнее задание (ИДЗ)**

#### *Варианты задания*

- 1) Напишите программу, которая организует хранение информации о студентах группы в файле. Позволяет вывести любую выбранную запись, добавить запись в файл, стереть запись из файла.
- 2) Напишите программу, которая организует хранение информации о книгах библиотеки в файле. Позволяет вывести любую выбранную запись, добавить запись в файл, стереть запись из файла.
- 3) Напишите программу, которая организует хранение информации о компьютерах в магазине в файле. Позволяет вывести любую выбранную запись, добавить запись в файл, стереть запись из файла.
- 4) Напишите программу, которая организует хранение информации о звездах солнечной системы в файле. Позволяет вывести любую выбранную запись, добавить запись в файл, стереть запись из файла.
- 5) Напишите программу, которая организует хранение информации о машинах салона в файле. Позволяет вывести любую выбранную запись, добавить запись в файл, стереть запись из файла.
- 6) Напишите программу, которая организует хранение информации о концертах музыкантов в файле. Позволяет вывести любую выбранную запись, добавить запись в файл, стереть запись из файла.
- 7) Напишите программу, которая организует хранение информации о театральных представлениях в файле. Позволяет вывести любую выбранную запись, добавить запись в файл, стереть запись из файла.
- 8) Напишите программу, которая организует хранение информации о животных зоопарка в файле. Позволяет вывести любую выбранную запись, добавить запись в файл, стереть запись из файла.
- 9) Напишите программу, которая организует хранение информации о моделях смартфона в файле. Позволяет вывести любую выбранную запись, добавить запись в файл, стереть запись из файла.
- 10) Напишите программу, которая организует хранение информации об ассортименте продуктового магазина в файле. Позволяет вывести любую выбранную запись, добавить запись в файл, стереть запись из файла.

Максимальная оценка – 10 баллов

**Таблица 12 – Критерии оценки сформированности компетенций**

| Код и наименование индикатора достижения компетенции (части компетенции)   | Критерии оценивания сформированности компетенции (части компетенции)   |   |   |
|--|--|---|---|
|  | на базовом уровне  | на повышенном уровне  |   |
|  | соответствует оценке «удовлетворительно»<br>50-64% от максимального балла  | соответствует оценке «хорошо»<br>65-85% от максимального балла  | соответствует оценке «отлично»<br>86-100% от максимального балла  |
| ИД-1пКос-2<br>Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент выполняет большую часть заданий, но неэффективно использует возможности файлового потока при работе с файлами, демонстрирует знание основного подхода к использованию файлов в создаваемых программах, способы осуществления параллельного и последовательного доступа к файлам, может создавать программы, использующие файлы для хранения и обработки информации, на базовом уровне может осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент выполняет все задания, но недостаточно эффективно использует возможности файлового потока при работе с файлами, демонстрирует знание и понимание основного подхода к использованию файлов в создаваемых программах, способы осуществления параллельного и последовательного доступа к файлам, способен создавать программы, использующие файлы для хранения и обработки информации, самостоятельно осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент полностью выполняет все задания, эффективно использует все возможности файлового потока при работе с текстовыми и бинарными файлами, демонстрирует глубокое знание и понимание основного подхода к использованию файлов в создаваемых программах, способы осуществления параллельного и последовательного доступа к файлам, с высоким уровнем самостоятельности способен создавать программы, использующие файлы для хранения и обработки информации, способен с высокой долей самостоятельности осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей |

### 3 ОПРЕДЕЛЕНИЕ РЕЗУЛЬТАТА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Формы промежуточной аттестации по дисциплине: *экзамен*.

#### ОЦЕНОЧНЫЕ МАТЕРИАЛЫ И СРЕДСТВА ДЛЯ ПРОВЕРКИ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

**ПКос-2. Способен осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей.**

#### Задания закрытого типа

*Выберите один правильный вариант ответа*

**1. Для хранения значений тока фидера нужно использовать переменную**

**типа:**

+double

int

string

char

**2. Для хранения названия подстанции нужно использовать переменную**

**типа:**

double

int

+string

char

#### Задания открытого типа

##### *Практико-ориентированные задания*

Задание 1

На подстанции для контроля параметров оборудования приборы измеряют активную и реактивную мощности. Написать на языке программирования C++ функцию, которая будет рассчитывать коэффициент мощности.

*Правильный ответ (решение):*

```
double PowerFactor(double active_power, double reactive_power)
{
    return active_power/sqrt(pow(active_power,2)+
        pow(reactive_power,2));
}
```

### Задание 2.

В ячейке КРУ вместо электромеханического амперметра поставили цифровой вместе с токовым шунтом. Написать на языке программирования C++ функцию для расчета сопротивления токового шунта. Входные данные: коэффициент шунтирования и входное сопротивление прибора.

*Правильный ответ (решение):*

```
double CurrentShunt(double n, double r_device)
{
    return r_device/(n-1);
}
```

### Задание 3.

В ячейке КРУ вместо электромеханического вольтметра поставили цифровой вместе с добавочным сопротивлением. Написать на языке программирования C++ функцию для расчета добавочного сопротивления. Входные данные: коэффициент расширения предела измерения и входное сопротивление прибора.

*Правильный ответ (решение):*

```
double AdditionalResistance(double m, double r_device)
{
    return r_device*(m-1);
}
```

### Задание 4.

Написать функцию, которая принимает два аргумента. Первый – ток фидера (в амперах), второй – верхний допустимый предел тока (в амперах). Если ток превышает предел, то функция в файл записывает величину тока и время (для хранения времени использовать переменную *time\_t*).

*Правильный ответ (решение):*

```
void CurrentWriter(double current, double max_current)
{
    if(current > max_current)
    {
        ofstream out("log");
        if(!out)
        {
            cerr<<" unable to open file";
            exit(1);
        }
        time_t accident_time = time(0);
        out<<current<< accident_time<<endl;
        out.close();
    }
}
```

### Задание 5.

Создайте класс, который содержит два закрытых поля: номер фидера и ток нагрузки. Значение полей должно устанавливаться при создании объекта класса. Опишите открытый метод, который выводит значения полей на экран через запятую.

*Правильный ответ (решение):*

```
class Feeder
{
    int number;
    double current;
public:
    Feeder(int n, double c): number(n), current(c) {}
    void display()
    {std::cout<<number<<“,“<<current;}
};
```

### Задание 6.

Создайте класс, который содержит закрытые поля: номер подстанции, название, мощность. Опишите открытые методы, которые устанавливают и возвращают значения полей класса.

*Правильный ответ (решение):*

```
class Substation
{
    int number;
    string name;
    double power;
private:
    void SetNumber(int n) {number=n;}
    void SetName(string n) {name=n;}
    void SetPower(double p) {power=p;}
    int GetNumber() {return number;}
    string GetName() {return name;}
    double GetPower() {return power;}
};
```

Окончательные результаты обучения (формирования компетенций) определяются посредством перевода баллов, набранных студентом в процессе освоения дисциплины, в оценки:

– базовый уровень сформированности компетенции считается достигнутым, если результат обучения соответствует оценке «удовлетворительно» (50-64 рейтинговых баллов);

– повышенный уровень сформированности компетенции считается достигнутым, если результат обучения соответствует оценкам «хорошо» (65-85 рейтинговых баллов) и «отлично» (86-100 рейтинговых баллов).



## 4 ПОРЯДОК ПРОВЕДЕНИЯ ПОВТОРНОЙ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Форма промежуточной аттестации по дисциплине *экзамен*.

Фонд оценочных средств для проведения повторной промежуточной аттестации формируется из числа оценочных средств по темам, которые не освоены студентом.

*Примечание:*

Дополнительные контрольные испытания проводятся для студентов, набравших менее **50 баллов** (в соответствии с «Положением о модульно-рейтинговой системе»).

**Таблица 13 – Критерии оценки сформированности компетенций**

| Код и наименование индикатора достижения компетенции (части компетенции)   | Критерии оценивания сформированности компетенции (части компетенции)  |
|--|---|
|  | на базовом уровне   |
|  | соответствует оценке «удовлетворительно»<br>50-64% от максимального балла   |
| ИД-1пКос-2<br>Осуществляет планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей | Студент демонстрирует знание синтаксиса языка программирования, средств разработки программы, операторов языка, процедурного подхода, структурированных типов, приемов использования адресной арифметики, объектно-ориентированного подхода к программированию, методов использования файлов, способов осуществления параллельного и последовательного доступа к файлам, способен использовать процедурный подход и создавать собственные функции, программы, использующие файлы для хранения и обработки информации, самостоятельно осуществлять планирование и контроль деятельности по техническому обслуживанию и ремонту оборудования подстанций электрических сетей |