

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Кузнецова Эмилия Васильевна
Должность: Исполнительный директор
Дата подписания: 23.11.2025 16:18:17
Уникальный программный ключ:
01e176f1d70ae109e92d86b7d8f33ec82fbb87d6

**ЧАСТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЕГИОНАЛЬНЫЙ ИНСТИТУТ БИЗНЕСА И УПРАВЛЕНИЯ»**

Рассмотрено и одобрено на заседании Учебно-
Методического совета
Протокол № 1 от 23 августа 2024 г.



УТВЕРДЖЕНО

Проректор по учебной работе

Ю.И. Паничкин

Личная подпись

инициалы, фамилия

«23» августа 2024 года

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

к рабочей программе дисциплины

«Алгоритмизация и программирование»

Направление подготовки

09.03.03 Прикладная информатика

Направленность
подготовки (профиль)

Прикладная информатика

Уровень программы

бакалавриат

Форма обучения

очно-заочная

Рязань 2024 г.

Фонд оценочных средств текущей и промежуточной аттестации по дисциплине «Иностранный язык»

Фонд оценочных средств является неотъемлемой частью рабочей программы дисциплины и основной образовательной программы.

Фонд оценочных средств представляет собой комплекс учебных заданий, предназначенных для измерения уровня достижений обучающимся установленных результатов обучения, и используется при проведении текущей и промежуточной аттестации (в период зачетно - экзаменационной сессии).

Цель ФОС – установление соответствия уровня подготовки обучающихся на данном этапе обучения требованиям рабочей программы дисциплины.

Основными задачами ФОС по учебной дисциплине являются:

- контроль достижений целей реализации ОП – формирование компетенций;
- контроль процесса приобретения обучающимся необходимых знаний, умений, навыков(владения/опыта деятельности) и уровня сформированности компетенций;
- оценка достижений обучающегося;
- обеспечение соответствия результатов обучения задачам будущей профессиональной деятельности через совершенствование методов обучения в образовательном процессе.

1. Планируемые результаты обучения по дисциплине в рамках планируемых результатов освоения основной образовательной программы. Перечень компетенций в процессе освоения образовательной программы.

Дисциплина «Алгоритмизация и программирование» обеспечивает освоение следующих компетенций с учетом этапа освоения:

Код компетенции	Наименование компетенции
ОПК-7	Способен разрабатывать алгоритмы и программы, пригодные для практического применения

Разработка алгоритмов решения задач	Структуры алгоритмов. Алгоритмы линейной структуры. Алгоритмы разветвляющейся структуры. Алгоритмы циклической структуры. Вычисления в цикле с несколькими одновременно изменяющимися параметрами.	О Т	ОПК-7
Основы программирования на языке C++.	Типы данных. Литералы. Переменные. Выражения и операции. Арифметические операции. Логические операции и операции сравнения. Линейная программа. Подключение библиотек. Поточковый ввод и вывод. Математические операции и функции. Разветвляющаяся программа. Оператор IF. Оператор SWITCH. Программа с циклической структурой. Оператор FOR. Операторы DO и WHILE.	О Т	ОПК-7
Указатели.	Указатель (pointer) в C++. Оператор получения адреса. Адрес переменной. Переменная-	О Т	ОПК-7

	указатель.		
Массивы и строки.	Многомерные массивы. Массив и указатель. Динамический массив. Массив типа CHAR. Тип данных STRING.	O T	ОПК-7
Функции	Аргументы и тип функции. Прототип функции. Рекурсия. Передача массивов как аргументов. Область видимости и время жизни переменных. Локальные и глобальные переменные.	O T	ОПК-7
Структуры..	Определения структур. Доступ к полям структуры. Вложенные структуры. Перечисления.	O T	ОПК-7
Объекты классы.	Простой класс. Объекты программы и объекты реального мира. Конструкторы. Структуры. Классы, объекты и память.	O T	ОПК-7
Перегрузка операций.	Перегрузка унарных операций. Перегрузка бинарных операций. Преобразование типов.	O T	ОПК-7
Наследование.	Базовый и производный классы. Конструкторы производного класса. Иерархия классов	O T	ОПК-7
Указатели. Управление памятью.	Адреса и указатели. Управление памятью. Связный список. Указатели на объекты и указатели.	O T	ОПК-7
Виртуальные функции.	Виртуальные функции. Дружественные функции. Статические функции.	O T	ОПК-7
Потоки и файлы	Потоковые классы. Поточковый ввод/вывод. Указатели файлов. Файловый ввод/вывод.	O T	ОПК-7

2.Соответствие уровня освоения компетенции планируемым результатам обучения и критериям их оценивания

Код компетенции	Наименование компетенции
ОПК-7	Способен разрабатывать алгоритмы и программы, пригодные для практического применения

Показатель оценивания/индикаторы	Критерии оценивания			
	2	3	4	5

Показатель оценивания/индикаторы	Критерии оценивания			
	2	3	4	5
Знает	<p>Студент не способен самостоятельно материале дисциплины. Не знает язык программирования, операционные системы и оболочки, современную программную среду разработки программного обеспечения - MS VisualStudio 2017. Показывает отсутствие знаний основ алгоритмизации и программирования, разработки алгоритмов различной структуры, основы построения программ на языке высокого уровня C++.</p>	<p>Студент усвоил основное содержание материала дисциплины, но имеет пробелы в усвоении материала. Имеет несистематизированные знания о языках программирования, операционные программирования, разработки алгоритмов различной структуры, основы построения программ на языке высокого уровня C++.</p>	<p>Студент самостоятельно выделять главные положения в изученном материале. Знает основы программирования, операционные системы и оболочки, современную программную среду разработки программного обеспечения. -MS VisualStudio 2017. Показывает глубокое знание основ алгоритмизации и программирования, разработки алгоритмов различной структуры, основы построения программ на языке высокого уровня C++.</p>	<p>Студент знает, понимает, выделяет главные положения в изученном материале и способен дать краткую характеристику основным идеям проработанного материала дисциплины. Знает основы программирования, операционные системы и оболочки, современную программную среду разработки программного обеспечения. - MS VisualStudio 2017. Показывает глубокое знание основ алгоритмизации и программирования, разработки алгоритмов различной структуры, основы построения программ на языке высокого уровня C++.</p>

Показатель оценивания/индикаторы	Критерии оценивания			
	2	3	4	5
Умеет	Студент не умеет применять языки программирования и, современную программную среду разработки программного обеспечения. - MS VisualStudio 2017. Не умеет разрабатывать алгоритмы прикладных задач различных классов различной структуры, на языке высокого уровня C++.	Студент умеет последовательно применять языки программирования и, современную программную среду разработки программного обеспечения. - MS VisualStudio 2017. Умеет разрабатывать алгоритмы прикладных задач различных классов различной структуры, на языке высокого уровня C++.	Студент умеет применять языки программирования и, современную программную среду разработки программного обеспечения MS VisualStudio 2017. Умеет разрабатывать алгоритмы прикладных задач различных классов структуры, на языке высокого уровня C++.	Студент умеет применять языки программирования и, современную программную среду разработки программного обеспечения. - MS VisualStudio 2017. Умеет разрабатывать алгоритмы прикладных задач различных классов различной структуры, на языке высокого уровня C++.
Владеет	Студент не владеет навыками программирования на языке высокого уровня C++ , отладки и тестирования прототипов программно-технических задач в среде разработки программного обеспечения. - MS VisualStudio 2017.	Студент владеет основными навыками программирования на языке высокого уровня C++, испытывает затруднения при отладке и тестировании прототипов программно-технических задач в среде разработки программного обеспечения. - MS VisualStudio 2017.	Студент владеет навыками программирования на языке высокого уровня C++, Допускает ошибки при отладке и тестировании задач в среде разработки программного обеспечения. - MS VisualStudio 2017.	Студент владеет навыками программирования на языке высокого уровня C++ , отладки и тестирования Разработки программного обеспечения. - MS VisualStudio 2017.

3. Фонд оценочных средств и материалы текущего контроля успеваемости обучающихся и промежуточной аттестации по дисциплине

3.1. В ходе реализации дисциплины «Алгоритмизация и программирование» используются следующие формы текущего контроля успеваемости обучающихся: опрос, тестирование, и т.д.

3.2. Преподаватель при текущем контроле успеваемости, оценивает уровень подготовленности обучающихся к занятию по следующим показателям:

- устные (письменные) ответы на вопросы преподавателя по теме занятия;
- количество правильных ответов при тестировании;
- по сформированности собственных суждений основанных на значимых фактах и практических результатах отраженных в реферате, эссе;
- аргументированности, актуальности, новизне содержания доклада;
- по точному выполнению целей и задач контрольной работы.

Детализация баллов и критерии оценки текущего контроля успеваемости утверждается на заседании кафедры.

3.2.1. Вопросы для подготовки к опросу по всем изучаемым тема дисциплины:

Задания в форме устного опроса

Семестр 1

1. Понятие алгоритма.
2. Структуры алгоритмов.
3. Алгоритмы линейной структуры.
4. Алгоритмы разветвляющейся структуры.
5. Алгоритмы с циклической структурой.
6. Вычисление суммы членов бесконечного ряда.
7. Вычисление полинома.
8. Нахождение наименьшего и наибольшего значений.
9. Алгоритм со структурой вложенных циклов.
10. Типы данных.
11. Литералы.
12. Переменные.
13. Выражения и операции.
14. Арифметические операции.
15. Логические операции и операции сравнения.
16. Линейная программа.
17. Подключение библиотек.
18. Поточковый ввод и вывод.
19. Математические операции и функции.
20. Разветвляющаяся программа.
21. Оператор IF.
22. Оператор SWITCH.
23. Программа с циклической структурой.
24. Оператор FOR.
25. Операторы DO и WHILE.
26. Указатель (pointer) в C++.
27. Оператор получения адреса.
28. Адрес переменной.

29. Переменная-указатель.

Семестр 2

1. Определение массивов.
2. Инициализация массивов.
3. Одномерные массивы.
4. Многомерные массивы.
5. Массив и указатель.
6. Динамический массив.
7. Массивы объектов.
8. Массивы строк
9. Массив типа CHAR.
10. Тип данных STRING.
11. Аргументы и тип функции.
12. Прототип функции.
13. Простые функции.
14. Передача аргументов в функцию.
15. Перегруженные функции.
16. Рекурсия.
17. Встраиваемые функции.
18. Ссылки на аргументы.
19. Передача массивов как аргументов.
20. Область видимости и время жизни переменных.
21. Класс памяти.
22. Локальные и глобальные переменные.
23. Определения структур.
24. Доступ к полям структуры.
25. Простая структура.
26. Вложенные структуры.
27. Перечисления.

Семестр 3

1. Объекты и классы
2. Простой класс.
3. Объекты программы и объекты реального мира.
4. Конструкторы.
5. Структуры.
6. Классы, объекты и память.
7. Перегрузка операций.
8. Перегрузка унарных операций.
9. Перегрузка бинарных операций.
10. Преобразование типов.
11. Наследование.
12. Базовый класс.
13. Производный класс.
14. Конструкторы производного класса.
15. Иерархия классов.

Семестр 4

1. Адреса и указатели.
2. Управление памятью
3. Связный список.

4. Указатели на объекты и указатели.
5. Виртуальные функции.
6. Дружественные функции.
7. Статические функции.
8. Динамическая информация о типах.
9. Потоки и файлы.
10. Потокосовые классы.
11. Потокосовый ввод/вывод.
12. Ошибки потока.
13. Указатели файлов.
14. Файловый ввод/вывод.
15. Обработка ошибок ввода/вывода.

Контролируемые компетенции: ОПК-7.

Устный (письменный) опрос проводится в течение установленного времени преподавателем. Опрашиваются все обучающиеся группы. За опрос выставляется оценка до 10 баллов. Набранные баллы являются рейтинг-баллами.

Рейтинг-баллы	Аттестационная оценка обучающегося по дисциплине учебного плана в национальной системе оценивания
8-10	отлично
6-7	хорошо
4-5	удовлетворительно
0-3	неудовлетворительно

При оценивании учитывается:

1. Целостность, правильность и полнота ответов
2. В ответе приводятся примеры из практики, даты, Ф.И.О. авторов
3. Применяются профессиональные термины и определения

Процедура оценки опроса:

1. Если ответ удовлетворяет 3-м условиям – 8-10 баллов.
2. Если ответ удовлетворяет 2-м условиям – 6-7 баллов.
3. Если ответ удовлетворяет 1-му условию – 4-5 баллов.
4. Если ответ не удовлетворяет ни одному условию – 0-3

5. 2.2. Тестовые задания для проведения тестирования:

Тестовые задания

Семестр 2

1. Укажите оператор выбора в языке C ++.
CASE
choice
switch ... case default
2. Для чего предназначен оператор continue в языке C ++?
Пропускает остаток тела цикла и переходит к следующей итерации.
Пропускает цикл и переходит к следующему оператору в теле программы.

Определяет условие продолжения цикла.

Продолжает выполнение текущей итерации цикла.

3. Значение переменной number не лежит между 3 и 6. Укажите правильный вариант записи данного утверждения на языке C ++.

number > 3 && number < 6

!(Number < 3 && number < 6)

!(Number < 6 || number > 3)

number < 3 || number > 6

4. Что выполняет операция ++ в языке C ++?

Уменьшает значение операнда на единицу.

Уменьшает значение операнда на два.

Увеличивает значение операнда на два.

Увеличивает значение операнда на единицу.

5. В программе на языке C ++ есть два объявления переменных int qwerty; int QWERTY; Какое из утверждений верно?

Такие имена переменных недопустимы.

Объявления правильные.

Такие объявления недопустимы, потому что мы пытаемся создать две переменные с одинаковыми идентификаторами.

Переменные описываются не по такому принципу.

6. Укажите запись экранированного символа языке C ++. 'F' «\022»

'Ю' '\t'

7. Зачем в C ++ используют оператор return?

Чтобы задержать работу программы.

Функция, в которой он содержится, завершает свое выполнение и управление

возвращается в то место программы, из которого вызывалась данная функция. +тобы организовать цикл.

+тобы ввести в программу новые значения.

8. В программе на языке C ++ объявлены такие переменные int x, y; Выражение позволяет вычислить остаток от деления этих переменных?

x % y

x div y

x mod y x /

y

9. Что выведет следующая программа?

```
#include <iostream.h>
```

```
int main() {
```

```
    int l_i ;
```

```
    for( l_i = 0; l_i < 9; l_i++)
```

```
        cout << l_i +1;
```

```
    return 1;
```

```
}
```

цифры от 0 до 8

программа не будет построена из-за ошибок

цифры от 1 до 9

10. Для того чтобы вывести символ новой строки, надо:

воспользоваться специальным манипулятором endl
при выводе строки символов перевод строки добавляется автоматически закончить
оператор точкой с запятой

11. Функция вычисляет произведение двух чисел. Исходные данные вводятся с клавиатуры. Какие проверки целесообразно ввести в программе?

проверка, что исходные данные являются числами и эти числа больше нуля
проверка, что исходные данные являются числами
проверка исходных данных на равенство нулю
проверки не нужны, все возможные ошибки отловит компилятор

12. Какие компоненты не входят в интегрированную среду программирования
- компилятор
 - текстовый редактор
 - отладчик переводчик

13. Укажите все ключевые слова в приведенном примере

```
int calc(int a, int b, bool f)
```

```
{  
    if (f==1)  
        return a+b;  
    else  
        return a*b;  
}
```

int, bool, if, else, return
int, calc, bool, return, if,
else int, if, else, return

14. В чем различие использования следующих выражений #include <...> и #include «...»

в различии использования заголовочных и исходных
файлов нет различий
различие заключается в методе поиска препроцессором включаемого файла

15. Чему будет равен результат вычисления выражения: int d=5; bool b = true, c; c = (!b||(d>3));

true
Ошибка компилятора
false

16. Если после выражения стоит точка с запятой, то
выражение вычисляется только если первой стоит операция присваивания это
оператор-выражение, действие которого заключается в вычислении выражения
выражение вычисляется, а его значение запоминается в специальной переменной,
которую можно использовать в следующем операторе

17. Отметьте истинные высказывания:

переменная инициализируется, потом объявляется переменная объявляется, потом
изменяется
переменная объявляется, потом инициализируется и изменяется

18. В каком случае программа выведет строку на консоль

```
cout < "Hello, world!" < endl;  
cout << "Hello, world!" << endl; cout >> "Hello, world!" >> endl;
```

19. Какой результат будет у следующего выражения?

```
int m = 1, n=2;  
double A = (double)m/n;  
cout << A;
```

0

1 ошибка компиляции 0.5

20. Процесс компиляции программы

приводит программы к единообразному внешнему виду переводит исходный текст в исполняемый файл для языка Си++ необязателен

21. +ему равно значение выражения

!((1 || 0) && 0) ?

ошибка компиляции

0

1

22. Если int n=3, какой будет результат выполнения кода?

```
switch(n) {  
case 2: cout << "aaa"; break; case  
3: cout << "bbb"; break; default:  
cout << "vvv"; break; }
```

ошибка компилятора

vvv aaa bbb

неопределенное поведение

23. Что будет выведено на экран в результате выполнения кода? int a=3;

```
if (a>1)  
cout << "1";  
else  
if(a>2)  
cout << "2";  
else  
if(a>3)  
cout << "3";
```

1

12

123

24. В каком случае компилятор не выдаст ошибку:

```
int int iCeloe;  
bool LD1LW;  
const float fL = 32; float e23 = 1; fL = e23;
```

25. Укажите правильный идентификатор для имени переменной:

FA_Ф12 int 2a _ri18

26. Если есть два объявления int qwerty; int QWERTY; какое из утверждений верно

такие имена переменных недопустимы объявления правильные

такие объявления недопустимы, так как мы пытаемся создать две одинаковые переменные

27. Битовой операцией (операцией с одним операндом) является &

|| + !=

28. чему равен результат вычисления выражения

$b - x * 3 + b$

при $x = 12$ и $b = 8$?

-4

-20

-124

29. Что является результатом компоновки программы?

набор заголовочных файлов с определением в них всех используемых функций

исполняемый файл или библиотека

заголовочный файл

30. Если имеется объявление «char ch1='a',ch2='b',ch3='c';»,

допустима ли запись «ch1=ch2+ch3»?

нет да

31. Чему равен результат вычисления выражения

$x + 3 * b + x$

при $x = 12$ и $b = 8$?

132

300

48

32. Если $i = 5$, какой будет результат вывода

do

{

cout << (++i)++ << " ";

}

while (i>=5 && i < 8);

6 7 8

6 8

6 7

6

33. Отметьте правильное объявление переменных:

float; float = y;

char float = 53.5;int x; int y; int X;

34. Какой из перечисленных типов является встроенным типом языка C++?

float

boolean

real

integer

35. Исходя из данного кода какое высказывание верно?

int main()

{

int a,b,c,d;

```

a=1; b=2;
c=a+b+p;
cout << p;
...
}

```

код не верен, потому что переменным c и d не присвоены значения

код не верен, потому что переменная p не объявлена

код верен, потому что по умолчанию все переменные имеют целочисленный тип

1. Какая из следующих функций считывает 100 символов из входного потока в строку x?

```

getline(x, 100, 'n');
cin.getline(x, 100, 'n');
read(x);
cin.getline(100, x, 'n');

```

2. Строковый типы данных в C++

строки в C++ представляются как массивы элементов типа char, заканчивающиеся терминатором строки - символом с нулевым значением ('/0').

строки в C++ представляются как массивы элементов типа char, заканчивающиеся терминатором строки - символом с нулевым значением ('').

строки в C++ представляются как массивы элементов типа char, заканчивающиеся терминатором строки - символом с нулевым значением '0'.

3. Код, указанный ниже объявляет массив ссылок. Правда это или нет? int main()

```

{
    int& x[50];

    return 0;
}

```

да

нет

4. Будет ли напечатано сообщение «не равны»? struct Foo

```
{
};

struct Bar
{
};

int main(int argc, char** argv)
{
    Foo* f = new Foo; Bar* b =
    new Bar;

    if ( f == b )
        std::cout << "равны" << std::endl;
    else
        std::cout << "не равны" << std::endl;

    return 0;
}
```

да нет

5. Программа напечатает строку / программист или нет? struct Foo

```
{
    int x;
    int y;
};

int main(int argc, char** argv)
{
    Foo f;

    if ( &f.x < &f.y )
    {
        std::cout << "/" программист" << std::endl;
    }

    return 0;
}
```

да нет

некорректное определение

6. Укажите правильное объявление указателя в C++

int &x; int x; int *x; ptr x;

7. Укажите правильное объявление массива

array an array[10];

int anarray[10];

int anarray;

anarray{10};

8. Строка Привет Мир будет показана на экране или нет?

```
int main(int argc, char** argv)
{
    int array[33];

    if ( &array[4] < &array[23] )
    {
        std::cout << "Привет мир" << std::endl;
    }

    return 0;
}
```

да

синтаксическая ошибка

нет

9. Что такое ссылка?

используется для переименования объектов

нет правильного ответа

ссылка является псевдонимом для объекта

оператор

10. Укажите зарезервированное ключевое слово для высвобождения выделенной памяти clear

remove

delete

free

11. Какой заголовочный файл необходимо подключить, чтобы вызвать функцию isalpha()?

cstring

conio.h

cctype

ifstream.h

12. Корректное выделение памяти

```
int a = new sizeof(int * 20);
int *a = new int[20];
int *a = new sizeof(int * 20);
int a = new int[20]; int
a = new int(20); int *a
= new 20;
```

13. Укажите статическую строку

'Статическая строка'

"Статическая строка"

char string[100];

14. Какая из следующих функций добавляет одну строку в конец другой?

stradd();

stringadd ();

strcat ();

Append ();

15. В какой из следующих записей используется операция взятия адреса?

```
&a;  
address(a);  
*a; a ;
```

16. В какой из следующих строк выполняется обращение к седьмому элементу массива, размер массива равен 10?

```
mas;  
mas(7);  
mas[6];  
mas[7];
```

17. В какой из следующих записей используется операция разыменования?

```
&a; a ;  
address(a);  
*a;
```

18. Какая из следующих записей возвращает значение переменной a?

```
*a;  
&a;  
val(a); a ;
```

Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие общий адрес. Доступ к отдельным элементам массива осуществляется по адресу и индексу

Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие имя.

Доступ к отдельным элементам массива осуществляется по имени массива и индексу

Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие имя.

Доступ к отдельным элементам массива осуществляется по имени массива и адресу

20. Какой стандартный код используется для Символьных данных типа char в C++?

```
Код ASCII  
Код ASCII  
Код UTF-8  
Код cp-1251
```

21. Укажите строку, которая возвращает адрес первого элемента в массиве arr?

```
&arr;  
arr;
```



```
arr[0];
arr[1];
```

22. Как правильно высвободить память, после выполнения этого кода?

```
char *a; a = new char[20];
```

```
delete a[];
delete a;
delete [] a;
```

23. После выполнения ряда операций с указателем, что будет выведено на экран?

```
int main(int argc, char** argv)
{
```

```
// предположим, int занимает 4 байта std::cout << sizeof(int) << std::endl;
```

```
int *x = new int;
```

```
// предположим адрес равен 0x60450000 std::cout << x << std::endl;
```

```
std::cout << x + 3 << std::endl;
```

```
return 0;
}
```

0x60450003

некорректное определение

нельзя заранее сказать, каково будет значение адреса

0x60450000

24. Словосочетание "Hello world!" может быть сохранено в символьном массиве размером n элементов. Укажите чему равно n?

11
13
12
10

25. Укажите корректное определение строковой переменной

```
string mystr[20];
string mystr;
string[20] mystr;
char mystr[20];
```

26. Каким символом завершается Си-строка?

"\0"

"\n"

27. Какое значение будет напечатано, в результате выполнения следующего кода?

```
#include <iostream>
```

```

int main()
{
    int sum = 0;

    int array[3][] = {{0, 1, 2}, {3, 4, 5}, {6, 7, 8}};

    for (int i = 0; i < 3 ; ++i)
    {
        for (int j = 2; j < 3 ; j++)
        {
            sum += array[i][j];
        }
    }

    std::cout << sum << std::endl;

    return 0;
}

```

15
синтаксическая ошибка

28. Объявлена переменная `char a`; Какое из следующих выражений не верно?

```

a = "3";
a = 3;
a = '3';

```

29. Укажите зарезервированное ключевое слово для динамического выделения памяти

```

new create
malloc value

```

30. Какой порядковый номер последнего элемента массива, размер массива 19?

18
порядковый номер определяется программистом 19

31. Какая из следующих функций сравнивает две строки?

```

stringcompare();
strcmp();
cmp();
compare();

```

32. В каком из вариантов ответов объявлен двумерный массив?

```

int array[20, 20];
char array[20];
int anarray[20][20];

array anarray[20][20];

```

33. Допустим, у нас есть код

```
char arr[8];  
cin >> arr;
```

И в массив arr мы попытались записать следующий набор символов Hello World. +то в действительности будет содержать массив arr?

```
Hello W  
Другой ответ  
Hello Wo Hello  
World Hello
```

Семестр 3

1. Можно ли перегрузить функцию main()?

да нет

2. Какая из переменных хранит количество аргументов, передаваемых в программу?

```
count arglen  
argc argv
```

3. Что будет напечатано на экране, после выполнения этого кода?

```
#include <iostream>  
  
int foo(int x, int y)  
{  
    return x+y;  
}  
  
double foo(double x, double y)  
{  
    return x+y;  
}  
  
int main(int argc, char** argv)  
{  
    double (*ptr)(int, int);  
  
    ptr = foo;  
  
    std::cout << ptr(3,8) << std::endl;  
  
    return 0;  
}
```

ошибка компиляции
11
8
3

4. Что будет напечатано на экране, после выполнения этого кода?

```
#include <ostream>
int foo(int y);
int foo(int x)
{
    return x+1;
}

int main(int argc, char** argv)
{
    int x = 3;
    int y = 6;

    std::cout << foo(x) << std::endl;

    return 0;
}
```

3
9
4
ошибка компиляции

5. Укажите тип возвращаемого значения следующей функции `int func(char x, float v, double t);`

double
float
int
char

6. Какой тип данных имеет переменная `ARGV`?

это не переменная
`char **`
`int char *`

7. Какие из следующих функций являются встроенными?

`inline void foo() {}`
`void foo() inline {}` `inline: void foo() {}` нет правильного ответа

8. Какая строка содержит зарезервированные слова языка программирования C++?

`if, else, for, while, do, switch, continue, break`
`char, int, float, double, short, long, unsigned, signed`
`defaulted, goto, return, extern, private, public, protected`
`sizeof, const, typedef, static, voided, enum, struct, union`

9. Для чего используются встроенные функции?

Для удаления ненужных функций
Чтобы уменьшить размер программы
Для упрощения файла с исходным кодом
Для увеличения скорости работы программы

10. Что значит ключевое слово `inline`?

Сообщает компилятору использовать функцию только в пределах одного файла с

исходным кодом

все вызовы встроенных функций заменяются кодом этой функции нет правильного ответа

все вызовы встроенных функции заменяются кодом этой функции

11. Какие из следующих утверждений верны?

компилятор может проигнорировать объявление встроенной функции.

встроенные функции не могут возвращать значения.

встроенные функции должны возвращать значение.

встроенные функции не должны содержать более 10 строк кода.

12. Что из нижеперечисленного не является прототипом функции?

double funct(char x)

char x();

int funct(char x, char y);

void funct();

13. Что будет напечатано на экране, после выполнения этого кода?

```
#include <iostream>
```

```
int foo(int x, int y)
```

```
{
```

```
    return x+y;
```

```
}
```

```
int foo(const int x, const int y)
```

```
{
```

```
    return x+y+1;
```

```
}
```

```
int main(int argc, char** argv)
```

```
{
```

```
    const int x = 3;
```

```
    const int y = 2;
```

```
    std::cout << foo(x,y) << std::endl;
```

```
    return 0;
```

```
}
```

6

3

5

ошибка компиляции

14. Укажите правильный вызов функции, предполагается, что функция была объявлена ранее.

funct;

```
int funct();;
funct x, y;
funct();
```

15. Что такое ARGV[0]?

ARGV[0] нигде не используется
первый аргумент, который передается в программу из командной строки массив

16. В каком порядке эти два параметра, объявлены в функции main?

Параметры: argc и argv

они не объявлены в main
Существует только один аргумент
количество аргументов, затем массив аргументов массив аргументов, затем количество элементов

17. Какое значение будет содержать переменная y?

```
const int x = 5;
int main(int argc, char** argv)
{
    int x[x];

    int y = sizeof(x) / sizeof(int);

    return 0;
}
```

5
20

18. Будет ли работать следующий код?

```
int x = 5;

template <typename T> class x

{
    T member;
};

int main(int argc, char** argv)
{
    class x<int> y;

    return 0;
}
```

да нет

19. Каков будет результат выполнения следующего кода?

```
int f(int a)
```

```

{
return ++a;
}
int f(unsigned int a)
{
return --a;
}
cout << f(5);

```

ошибка компиляции

6

5

4

20. Какое значение будет содержать локальная переменная x, в конце main?

```

int x = 5;
int main(int argc, char** argv)
{
    int x = x;
    return 0;
}

```

неопределенное

0

5

21. Выберите правильное (полное) определение функции.

- ```
void funct(int)
{
 cout << "Hello"
}
int funct(int x)
{
 return x = x + 1;
}
```
- ```
int funct();

void funct(x)
{
    cout << "Hello"
}
```

22. Можно ли гарантировать, что объявленная встроенная функция действительно является встроенной?

можно с уверенностью гарантировать, что объявленная вами функция как встроенная, действительно будет встроенной
 гарантировать невозможно, в каждом индивидуальном случае бывает по-разному

23. Возможна ли такая ситуация?

```
int x = 5;

class x
{
};

int main(int argc, char** argv)
{
    class x y;

    return 0;
}
```

нет да

Семестр 4

1. Правильное определение структуры в C++

```
struct a_struct {int a;};
struct {int a;}
struct a_struct {int a;}
struct a_struct int a;
```

2. Какой из следующих классов обрабатывает процесс записи в файл?

ofstream
другое
input_file ifstream

3. Какой заголовочный файл C++ содержит инструкции файлового ввода/вывода?

fstream
ifstream
ifstream iostream

4. Как организовать запись в файл?

```
a_file.printf("запись");
a_file="запись";
a_file.out("запись"); a_file
<< "запись ";
```

5. Какие из перечисленных констант определяют режим открытия файлов в C++?

ios_base::create
ios_base::application
ios_base::noreplace ios_base::trunc

6. Укажите правильный доступ к переменной структуры

b-var;


```
b.var;  
b->var;  
b>var;
```

7. Что означает константа `ios_base::ate`, передаваемая в конструктор, в качестве аргумента?

Открыть файл только для чтения.
Открыть файл, предварительно создав его.
Открыть файл, не создавая его.
При открытии переместить указатель в конец файла.

8. Какое значение будет напечатано?

```
#include <iostream>  
  
const int SIZE = 5; struct  
  
tester  
{  
    int array[::SIZE];  
  
    enum  
    {  
  
        SIZE = 3  
    };  
  
    void size()  
    {  
        std::cout << sizeof(array) / sizeof(int);  
    }  
};  
  
int main(int argc, char** argv)  
{ tester t; t.size();  
  return 0;  
}  
  
3  
заранее неизвестно  
5
```

9. Какое значение будет напечатано?

```
#include <iostream>  
  
const int x = 12;  
  
int main(int argc, char** argv)  
{  
    enum dog  
    {  
        x = x, y };  
}
```

```
std::cout << y << std::endl;

return 0;
}

13
12
Неопределенное
```

10. В каком из следующих вариантов ответов выполнен корректный доступ к переменной структуры, причём структура объявлена через указатель?

```
b-var;
b.var;
b->var;
b>var;
```

11. При определении структуры необходимо использовать следующее ключевое слово
re co rd o bj ec t st ru ct ur e st ru ct

12. Правильное объявление переменной, типа структуры foo

```
f o o v a r ; i n t f o o ; f o o ; struct foo;
```

13. Правильная конструкция определения класса в C++

```
Class < имя класса > { Private:
    <список объявлений данных-элементов и функций-элементов, скрытых от -
    доступ
- Public:
    <список прототипов функций-элементов общедоступного интерфейса> };

Class = < имя класса > { Private
    <список объявлений данных-элементов и функций -элементов, скрытых
от дост
Public
    <список прототипов функций-элементов общедоступного интерфейса>};

dass < имя класса >
{
private:
    <список объявлений данных-элементов и функций-элементов, скрытых от
досту
public:
    <список прототипов функций -элементов общедоступного интерфейс > };
```

14. Какое значение должен возвращать деструктор?

```
код состояния о правильном удалении
класса деструкторы не возвращают
значение
объект
класса.
указатель на класс
```

15. Может ли деструктор иметь аргументы?

```
да
```

нет

16. Понятие this в классе

Указатель this является скрытым аргументом метода, превращает функцию в область памяти только для чтения.

Объект this является аргументом метода другого класса, существует во всех методах и указывает на адрес: this -><объект>

Указатель this является скрытым аргументом метода, существует во всех методах объекта и указывает на его (объект) адрес: this ->< объект >

17. Назовите принципы объектно-ориентированного представления программных систем

абстрагирование; инкапсуляция; модульность; абстрагирование;
модульность; иерархическая организация абстрагирование; инкапсуляция;
модульность; иерархическая организация

18. Что такое деструктор?

Деструктор - это специальная функция -элемент, которая должна отслеживать данные в экземпляре класса в процессе работы

Деструктор - это функция, которая должна открывать динамическую область для экземпляра класса

Деструктор - это специальная функция-элемент, которая должна уничтожать экземпляр класса после завершения его работы

19. Назовите преимущества классов

все указанные ответы

удобный способ моделирования объектов реального мира инкапсуляция данных
удобство повторного использования кода

20. Какого спецификатора доступа в классах нет?

protected
public
hidden
private

21. Что такое объект?

Объект - это конкретное представление абстракции с характеристиками – модификатором, селектором, итератором.

Объекты это инструмент борьбы для борьбы со сложностью различных систем реальных сущностей с характеристиками – агрегация, зависимость, конкретизация.

Объект - это конкретное представление абстракции, которое обладает индивидуальностью, состоянием и поведением.

22. Что такое класс?

Класс - это копия характеристик объекта , которые разделяют одинаковые свойства, операции, отношения семантику (смысл).

Класс - это абстракция существенных характеристик объекта или это описание множества объектов, которые разделяют одинаковые свойства, операции, отношения семантику (смысл)

Класс - это абстракция существенных характеристик системы или это описание множества систем, которые имеют свойства с одним смыслом

23. Укажите корректное объявление класса

```
public class A { }  
object A { int x; };  
class B { }  
class A { int x; };
```

24. Какие функции есть у любого класса?

конструктор
нет таких
деструктор
конструктор и деструктор

25. Основные типы наследования в классах

Открытое наследование классов позволяет выполнить образование производного класса и объекта. Закрытое наследование классов позволяет выполнить образование

только производного класса. Множественное наследование - если у производного класса имеется несколько закрытых классов

Открытое наследование классов позволяет выполнить образование производного класса и объекта. Закрытое наследование классов позволяет выполнить образование только производного объекта. Множественное наследование - если у производного класса имеется несколько базовых классов

Открытое наследование классов позволяет выполнить образование производного класса и объекта. Закрытое наследование классов позволяет выполнить образование только производного класса. Множественное наследование - если у производного класса имеется несколько базовых классов

26. Какая функция выполняет начальную инициализацию данных в классе?

деструктор
нет правильного ответа
конструктор

Контролируемые компетенции: ОПК-7.

За выполнение контрольного теста выставляется оценка до 20 баллов. Набранные баллы являются рейтинг-баллами.

Параметры оценивания:

0-2 ошибки: «отлично» (18-20 баллов);

3-4 ошибки: «хорошо» (15-17 баллов);

5-6 ошибки: «удовлетворительно» (10-14 баллов)

7 и более ошибок: «неудовлетворительно» (1 -9 баллов)

Рейтинг-баллы	Аттестационная оценка обучающегося по дисциплине учебного плана в национальной системе оценивания
18-20	Отлично
15-17	Хорошо
10-14	Удовлетворительно
1-9	Неудовлетворительно

26.4.4. Тематика контрольных работ (не предусмотрена)

Контрольная работа предполагает выработку умений обучающимся показать глубокое знание теории предмета; на основе материала, установить и проанализировать следственно-логические связи и продемонстрировать навыки практического применения теоретической информации изучаемой дисциплины. Написание контрольной работы требует формулирование цели и задачи всей работы, заключение или выводы следуют из поставленных целей и задач.

За контрольную работу выставляется оценка до 20 баллов. Набранные баллы являются рейтинг-баллами.

Критерии оценки контрольной работы:

1. Выполнение задания в срок. Соответствие содержания заявленной теме;
2. Самостоятельность в выполнении работы, точность и полнота изложенного материала.
3. Логическое изложение материала. Соблюдение требований к оформлению работы.

Процедура оценки контрольной работы:

1. Если ответ удовлетворяет 3-м условиям – 18-20 баллов.
2. Если ответ удовлетворяет 2-м условиям – 15-17 баллов.
3. Если ответ удовлетворяет 1-му условию – 10-14 баллов.
4. Если ответ не удовлетворяет ни одному условию – 1-9

Рейтинг-баллы	Аттестационная оценка студента по дисциплине учебного плана в национальной системе оценивания
18-20	Отлично
15-17	Хорошо
10-14	Удовлетворительно
1-9	Неудовлетворительно

5. Форма и средства (методы) проведения промежуточной аттестации

5.1. Промежуточный контроль: зачет, экзамен (рейтинговая система)

Зачет и экзамен проводится в устной форме. Время, отведенное на подготовку вопросов зачета, экзамена составляет 15 мин. По рейтинговой системе оценки, формы контроля оцениваются отдельно. Зачёт и экзамен составляет от 0 до 20 баллов. Допуск к зачету и экзамену составляет 45 баллов.

Вопросы к зачету с оценкой и к экзамену

Второй семестр

1. Структуры алгоритмов.
2. Алгоритмы линейной структуры.
3. Алгоритмы разветвляющейся структуры.
4. Алгоритмы циклической структуры.
5. Вычисления в цикле с несколькими одновременно изменяющимися параметрами.
6. Типы данных.
7. Литералы. Переменные.

8. Выражения и операции.
9. Арифметические операции.
10. Логические операции и операции сравнения.
11. Линейная программа.
12. Подключение библиотек.
13. Поточковый ввод и вывод.
14. Математические операции и функции.
15. Разветвляющаяся программа.

16. Оператор IF.
17. Оператор SWITCH.
18. Программа с циклической
19. структурой.
20. Оператор FOR.
21. Операторы DO и WHILE.
22. Указатель (pointer) в C++.
23. Оператор получения адреса.
24. Адрес переменной.

Третий семестр

1. Структуры алгоритмов. Алгоритмы линейной структуры.
2. Алгоритмы разветвляющейся структуры. Алгоритмы циклической структуры.
3. Вычисления в цикле с несколькими одновременно изменяющимися параметрами.
4. Типы данных. Литералы. Переменные.
5. Выражения и операции. Арифметические операции. Логические операции и операции сравнения.
6. Линейная программа.
7. Подключение библиотек.
8. Поточковый ввод и вывод.
9. Математические операции и функции.
10. Разветвляющаяся программа.
11. Оператор IF. Оператор SWITCH.
12. Программа с циклической структурой.
13. Оператор FOR. Операторы DO и WHILE.
14. Указатель (pointer) в C++. Оператор получения адреса.
15. Адрес переменной.
16. Переменная-указатель.
17. Адреса и указатели.
18. Операции получения адреса.
19. Указатели- константы и указатели-переменные.
20. Определение массивов. Многомерные массивы.
21. Массивы объектов. Массивы строк.
22. Простые функции. Передача аргументов в функцию.
23. Ссылки на аргументы. Область видимости и время жизни переменных.
24. Класс памяти.
25. Структуры. Локальные и глобальные структур.
26. Доступ к полям структуры.
27. Вложенные структуры.
28. Перечисления
29. Простой класс.
30. Объекты программы и объекты реального мира.
31. Конструкторы.
32. Структуры. Классы, объекты и память.
33. Перегрузка унарных операций.
34. Перегрузка бинарных операций. Преобразование типов.

35. Базовый и производный классы.
36. Конструкторы производного класса.
37. Иерархия
38. Адреса и указатели.
39. Управление памятью.
40. Связный список.
41. Указатели на объекты и указатели.
42. Виртуальные функции.
43. Дружественные функции.
44. Статические функции.
45. Потокковые классы.
46. Потокковый ввод/вывод.
47. Указатели файлов.
48. Файловый ввод/вывод.

9.5. Вопросы к экзамену

1. Структуры алгоритмов. Алгоритмы линейной структуры.
2. Алгоритмы разветвляющейся структуры.
3. Алгоритмы циклической структуры.
4. Вычисления в цикле с несколькими одновременно изменяющимися параметрами.
5. Типы данных. Литералы. Переменные.
6. Выражения и операции.
7. Арифметические операции.
8. Логические операции и операции сравнения.
9. Линейная программа.
10. Подключение библиотек.
11. Потокковый ввод и вывод.
12. Математические операции и функции.
13. Разветвляющаяся программа.
14. Оператор IF. Оператор SWITCH.
15. Программа с циклической структурой.
16. Оператор FOR. Операторы DO и WHILE.
17. Указатель (pointer) в C++. Оператор получения адреса.
18. Адрес переменной.
19. Переменная-указатель.
20. Адреса и указатели.
21. Операции получения адреса.
22. Указатели- константы и указатели-переменные.
23. Определение массивов. Многомерные массивы.
24. Массивы объектов. Массивы строк.
25. Простые функции. Передача аргументов в функцию.
26. Ссылки на аргументы.
27. Область видимости и время жизни переменных.
28. Класс памяти.
29. Структуры. Локальные и глобальные структур.
30. Доступ к полям структуры.
31. Вложенные структуры.
32. Перечисления
33. Простой класс.
34. Объекты программы и объекты реального мира.
35. Конструкторы.
36. Структуры. Классы, объекты и память.
37. Перегрузка унарных операций.
38. Перегрузка бинарных операций.
39. Преобразование типов.

40. Базовый и производный классы.
41. Конструкторы производного класса.
42. Иерархия

Градация перевода рейтинговых баллов обучающихся в пятибалльную систему аттестационных оценок и систему аттестационных оценок ECTS.

Академический рейтинг обучающегося	Аттестационная оценка обучающегося по дисциплине учебного плана в национальной системе оценивания	Аттестационная оценка обучающегося по дисциплине учебного плана в системе ECTS
95-100	Отлично	+ A (excellent)
80-94		A (excellent)
75-79	Хорошо	+B (good)
70-74		B (good)
55-69	Удовлетворительно	C (satisfactory)
50-54		D (satisfactory)
45-49	Неудовлетворительно	E (satisfactory failed)
1-44		F (not rated)
0		N/A (not rated)

5. Практическая работа (практическая подготовка): проверка выполнения заданий по практической подготовке в профессиональной деятельности и самостоятельной работы на практических занятиях.

Практическое задание – это частично регламентированное задание по практической подготовке в профессиональной деятельности, имеющее алгоритмическое или нестандартное решение, позволяющее диагностировать умения, интегрировать знания различных научных областей в практическую подготовку связанную с профессиональной деятельности. Может выполняться в индивидуальном порядке или группой обучающихся.

Работа во время проведения практического занятия состоит из следующих элементов:

- консультирование обучающихся преподавателем с целью предоставления исчерпывающей информации, необходимой для самостоятельного выполнения предложенных преподавателем практических заданий и задач;
- самостоятельное выполнение практических заданий согласно обозначенной учебной программой тематики;
- ознакомление с инструктивными материалами с целью осознания задач практического занятия, техники безопасности при работе в аудитории.

Обработка, обобщение полученных результатов практической подготовки проводится обучающимися самостоятельно или под руководством преподавателя (в зависимости от степени сложности поставленных задач).

6. Примерные темы к курсовым работам(проектам)

Курсовая работа/проект – предусмотрена/не предусмотрена

7. Оценка компетенций (в целом)

Оценка компетенций (в целом) осуществляется по итогам суммирования текущих результатов обучающегося и промежуточной аттестации.

В оценке освоения компетенций (в целом) учитывают: полноту знания учебного материала по теме, степень активности обучающегося на занятиях в семестре; логичность

изложения материала; аргументированность ответа; уровень самостоятельного мышления, практической подготовки; умение связывать теоретические положения с практикой, в том числе и с будущей профессиональной деятельностью с промежуточной аттестации.