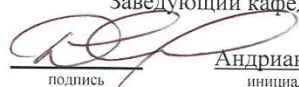


Приложение

Министерство образования и науки Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Кафедра ИС

УТВЕРЖДАЮ
Заведующий кафедрой ИС


подпись Андреанов Д. Е.
инициалы, фамилия

« 24 » 05 2016 г.

Основание:
решение кафедры ИС
от « 24 » 05 2016 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПРИ ИЗУЧЕНИИ УЧЕБНОЙ ДИСЦИПЛИНЫ**

Прикладное программное обеспечение
наименование дисциплины

09.03.03 Прикладная информатика
код и наименование направления подготовки

наименование профиля подготовки

бакалавриат
уровень высшего образования

Муром, 2016 г.

ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств (ФОС) для текущего контроля успеваемости и промежуточной аттестации по дисциплине «Прикладное программное обеспечение» разработан в соответствии с рабочей программой, входящей в ОПОП направления подготовки 09.03.03 Прикладная информатика.

№№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Решение задач на ЭВМ. Алгоритмы. Программное обеспечение.	ПК-2	тест
2	Арифметические и логические операции, оператор выбора if, ввод/вывод, приведение типов данных.	ПК-2	тест
3	Массивы и циклы	ПК-2	тест
4	Структуры и функции	ПК-2	тест
5	Строки, файлы и исключения	ПК-2	тест

Фонд оценочных средств по дисциплине «Прикладное программное обеспечение» предназначен для аттестации обучающихся на соответствие их персональных достижений поэтапным требованиям образовательной программы, в том числе рабочей программы дисциплины «Прикладное программное обеспечение», для оценивания результатов обучения: знаний, умений, владений и уровня приобретенных компетенций.

Фонд оценочных средств по дисциплине «Прикладное программное обеспечение» включает:

1. Оценочные средства для проведения текущего контроля успеваемости:

- комплект заданий репродуктивного уровня для выполнения на лабораторных и практических занятиях, позволяющих оценивать и диагностировать знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, распознавание объектов изучения в рамках определенного раздела дисциплины;
- тесты как система стандартизированных знаний, позволяющая провести процедуру измерения уровня знаний и умений обучающихся;

2. Оценочные средства для проведения промежуточной аттестации в форме:
итоговый тест для проведения экзамена

Перечень компетенций, формируемых в процессе изучения дисциплины «Прикладное программное обеспечение» при освоении образовательной программы по направлению подготовки 09.03.03 Прикладная информатика:

ПК-2: способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение		
Знать	Уметь	Владеть
синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования	применять выбранные языки программирования для написания программного кода	-

В результате освоения дисциплины «Прикладное программное обеспечение» формируется компетенция ПК-2: способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение.

Показатели, критерии и шкала оценивания компетенций текущего контроля знаний по учебной дисциплине «Прикладное программное обеспечение»

Текущий контроль знаний, согласно положению о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся (далее Положение) в рамках изучения дисциплины «Прикладное программное обеспечение» предполагает тестирование и выполнение заданий по лабораторным работам.

Регламент проведения и оценивание тестирования студентов

В целях закрепления практического материала и углубления теоретических знаний по разделам дисциплины «Прикладное программное обеспечение» предполагается выполнение тестирования студентов, что позволяет углубить процесс познания, раскрыть понимание прикладной значимости осваиваемой дисциплины.

Регламент проведения мероприятия

№	Вид работы	Продолжительность
1.	Предел длительности решения задачи	40 мин.
	Итого (в расчете на тест)	40 мин.

Критерии оценки тестирования студентов

Оценка выполнения тестов	Критерии оценки
<i>1 балл за правильный ответ на 1 вопрос</i>	<i>правильно выбранный вариант ответа (в случае закрытого теста), правильно вписанный ответ (в случае открытого теста)</i>

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ ЗНАНИЙ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ «Прикладное программное обеспечение»

1. Язык программирования C++ разработал

1. Дональд Кнут
2. Кен Томпсон
3. Никлаус Вирт
4. Бьерн Страуструп

2. Какому зарезервированному слову программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением?

1. all
2. contingency
3. default
4. other

3. Какие среды программирования (IDE) предназначены для разработки программных средств?

1. MVS, Code::Blocks, QT Creator, AutoCAD, Eclipse
2. MVS, Code::Blocks, QT Creator, RAD Studio, MathCAD
3. MVS, NetBeans, QT Creator, RAD Studio, Dev-C++

4. Укажите правильную форму записи цикла do while

1.
// форма записи оператора цикла do while:
do // начало цикла do while
{
/*блок операторов*/;
}
while (/*условие выполнения цикла*/) // конец цикла do while

2.
// форма записи оператора цикла do while:
do // начало цикла do while
{
/*блок операторов*/;
}
while (/*условие выполнения цикла*/); // конец цикла do while

3.
// форма записи оператора цикла do while:
do // начало цикла do while
{
/*блок операторов*/;
}
while {/*условие выполнения цикла*/} // конец цикла do while

5. Укажите объектно-ориентированный язык программирования

1. Eiffel
2. Java
3. Все варианты ответов
4. C++

6. Цикл с постусловием?

1. for
2. do while
3. while

7. Цикл с предусловием?

1. do while
2. for
3. while

8. Программа, переводящая входную программу на исходном языке в эквивалентную ей выходную программу на результирующем языке, называется:

1. компилятор
2. транслятор
3. сканер
4. интерпретатор

9. Простые типы данных в C++.

1. целые – int, вещественные – float или real, символьные – char
2. целые – bool, вещественные – float или double, символьные – string
3. целые – int, вещественные – float или double, символьные – string
4. целые – int, вещественные – float или double, символьные – char

10. Структура объявления переменных в C++

1. [=], <идент. 2>, ...;
2. [=]; <идент. 2>, ...;
3. [:=], <идент. 2>, ...;
4. [==]; <идент. 2>, ...;

11. Какая из следующих записей - правильный комментарий в C++?

1. /* комментарий */
2. {комментарий}
3. */ Комментарии */
4. ** Комментарий **

12. Какой оператор не допускает перехода от одного константного выражения к другому?

1. точка с запятой
2. Stop;
3. end;

4. break;

13. Какие служебные символы используются для обозначения начала и конца блока кода?

1. begin end
2. { }
3. < >
4. ()

14. Чтобы подключить заголовочный файл в программу на C++, например `iostream` необходимо написать:

1. include `#iostream,h;`
2. #include <> с `iostream.h` внутри скобок
3. #include <> с `iostream` внутри скобок
4. include (`iostreamh`)

15. Тело любого цикла выполняется до тех пор, пока его условие ...

1. ложно
2. истинно
3. у цикла нет условия

16. Общий формат оператора множественного выбора - switch

1.

```
switch (switch_expression)
{
    case constant1: statement1; [break;]
    case constant2: statement2; [break;]
    case constantN: statementN; [break;]
    [default: statement N+1;]
}
```
2.

```
switch (switch_expression)
{
    case constant1: statement1; [break;]
    case constant2: statement2; [break;]
    case constantN: statementN; [break;]
    [else: statement N+1;]
}
```
3.

```
switch (switch_expression)
{
    case constant1, case constant2: statement1; [break;]
    case constantN: statementN; [break;]
    [default: statement N+1;]
}
```

17. Какую функцию должны содержать все программы на C++?

1. start()
2. main()
3. system()
4. program()

18. Какой из ниже перечисленных операторов, не является циклом в C++?

1. repeat until
2. while
3. do while
4. for

19. Какой из следующих операторов - оператор сравнения двух переменных?

1. :=
2. =
3. equal
4. ==

20. Чему будет равна переменная a, после выполнения этого кода `int a; for(a = 0; a < 10; a++) { }`?

1. 10
2. 9
3. 1

21. Каков результат работы следующего фрагмента кода?

```
int x = 0;
```

```
switch(x)
```

```
{
```

```
    case 1: cout << "Один";
```

```
    case 0: cout << "Ноль";
```

```
    case 2: cout << "Привет мир";
```

```
}
```

1. Один
2. Ноль
3. НольПривет мир
4. Привет мир

22. Какой служебный знак ставится после оператора case ?

1. .
2. :
3. ;

4. -

23. Выберите правильный вариант объявления константной переменной в C++, где `type` - тип данных в C++ `variable` - имя переменной `value` - константное значение

1. `const type variable = value;`
2. `const type variable := value;`
3. `const variable = value;`

24. Какими знаками заканчивается большинство строк кода в Си++?

1. `,` (запятая)
2. `.` (точка)
3. `;` (точка с запятой)
4. `:` (двоеточие)

25. Какой из перечисленных типов данных не является типом данных в C++?

1. `double`
2. `real`
3. `int`
4. `float`

26. Какое значение, по умолчанию, возвращает программа операционной системе в случае успешного завершения?

1. 0
2. -1
3. Программа не возвращает значение.
4. 1

27. Название C++ предложил

1. Рик Масситти
2. Кэн Томпсон
3. Бьерн Страуструп
4. Дональд Кнут

28. До каких пор будут выполняться операторы в теле цикла `while (x < 100)`?

1. Пока `x` строго меньше ста
2. Пока `x` меньше или равен ста
3. Пока `x` равен ста
4. Пока `x` больше ста

29. Укажите правильное определение функции `main` в соответствии со спецификацией стандарта ANSI

1. `int main(void)`
2. `int main()`
3. `void main()`
4. `void main(void)`

30. Какие преобразования типов данных не возможны без потери данных?

1. char to float
2. все перечисленные преобразования не возможны
3. float to int
4. int to float

Регламент проведения и оценивание лабораторных работ

В целях закрепления практического материала и углубления теоретических знаний по разделам дисциплины «Прикладное программное обеспечение» предполагается выполнение лабораторных работ, что позволяет углубить процесс познания, раскрыть понимание прикладной значимости осваиваемой дисциплины.

Регламент проведения мероприятия

№	Вид работы	Продолжительность
1.	Предел длительности лабораторной работы	170 мин.
2.	Защита отчета	10 мин.
	Итого (в расчете на одну лабораторную работу)	180 мин.

Критерии оценки лабораторных работ

Оценка	Критерии оценивания
5 баллов	Задание выполнено полностью, в представленном отчете обоснованно получено правильное выполненное задание.
4 балла	Задание выполнено полностью, но нет достаточного обоснования или при верном решении допущена незначительная ошибка, не влияющая на правильную последовательность рассуждений.
3 балла	Задания выполнены частично.
2 балла	Задание не выполнено.

Общее распределение баллов текущего контроля по видам учебных работ для студентов (в соответствии с Положением)

Рейтинг-контроль 1	тест	до 15
Рейтинг-контроль 2	тест	до 15
Рейтинг-контроль 3	тест	до 15

Посещение занятий студентом		до 5
Дополнительные баллы (бонусы)		до 5
Выполнение семестрового плана самостоятельной работы		до 5

Показатели, критерии и шкала оценивания компетенций промежуточной аттестации знаний по учебной дисциплине «Прикладное программное обеспечение»

На основе перечня вопросов к тестированию программным комплексом информационно-образовательного портала МИ ВлГУ формируются в автоматическом режиме тестовые задания для студентов: 8 вопросов из блока 1, 7 вопросов из блока 2. Программный комплекс формирует индивидуальные задания для каждого зарегистрированного в системе студента и устанавливает время прохождения тестирования. Результатом тестирования является балл, рассчитанный на основе количества правильных ответов. С учетом индивидуального семестрового рейтинга студента формируется итоговый балл по курсу.

Максимальное количество баллов, которое студент может получить на экзамене, в соответствии с Положением составляет 40 баллов.

Оценка в баллах	Критерии оценивания компетенций
30-40 баллов	Студент глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач, подтверждает полное освоение компетенций, предусмотренных программой экзамена.
20-29 баллов	Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения, допуская некоторые неточности; демонстрирует хороший уровень освоения материала, информационной и коммуникативной культуры и в целом подтверждает освоение компетенций, предусмотренных программой экзамена.

10-19 баллов	Студент показывает знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, в целом, не препятствует усвоению последующего программного материала, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ, подтверждает освоение компетенций, предусмотренных программой экзамена на минимально допустимом уровне.
Менее 10 баллов	Студент не знает значительной части программного материала (менее 50% правильно выполненных заданий от общего объема работы), допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, не подтверждает освоение компетенций, предусмотренных программой экзамена.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ «Прикладное программное обеспечение»

Блок 1 (знать):

1. Язык программирования C++ разработал

1. Дональд Кнут
2. Кен Томпсон
3. Никлаус Вирт
4. Бьерн Страуструп

2. Какому зарезервированному слову программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением?

1. all
2. contingency
3. default
4. other

3. Какие среды программирования (IDE) предназначены для разработки программных средств?

1. MVS, Code::Blocks, QT Creator, AutoCAD, Eclipse
2. MVS, Code::Blocks, QT Creator, RAD Studio, MathCAD
3. MVS, NetBeans, QT Creator, RAD Studio, Dev-C++

4. Укажите правильную форму записи цикла do while

1.


```
// форма записи оператора цикла do while:
do // начало цикла do while
{
/*блок операторов*/;
```

```
}  
while (/*условие выполнения цикла*/) // конец цикла do while  
2.
```

```
    // форма записи оператора цикла do while:  
do // начало цикла do while  
{  
/*блок операторов*/;  
}  
while (/*условие выполнения цикла*/); // конец цикла do while  
3.
```

```
    // форма записи оператора цикла do while:  
do // начало цикла do while  
{  
/*блок операторов*/;  
}  
while {/*условие выполнения цикла*/} // конец цикла do while
```

5. Укажите объектно-ориентированный язык программирования

1. Eiffel
2. Java
3. Все варианты ответов
4. C++

6. Цикл с постусловием?

1. for
2. do while
3. while

7. Цикл с предусловием?

1. do while
2. for
3. while

8. Программа, переводящая входную программу на исходном языке в эквивалентную ей выходную программу на результирующем языке, называется:

1. компилятор
2. транслятор
3. сканер
4. интерпретатор

9. Простые типы данных в C++.

1. целые – int, вещественные – float или real, символьные – char
2. целые – bool, вещественные – float или double, символьные – string
3. целые – int, вещественные – float или double, символьные – string
4. целые – int, вещественные – float или double, символьные – char

10. Структура объявления переменных в C++

1. [=], <идент. 2>, ...;
2. [=]; <идент. 2>, ...;
3. [:=], <идент. 2>, ...;
4. [==]; <идент. 2>, ...;

11. Какая из следующих записей - правильный комментарий в C++?

1. /* комментарий */
2. {комментарий}
3. */ Комментарии */
4. ** Комментарий **

12. Какой оператор не допускает перехода от одного константного выражения к другому?

1. точка с запятой
2. Stop;
3. end;
4. break;

13. Какие служебные символы используются для обозначения начала и конца блока кода?

1. begin end
2. { }
3. < >
4. ()

14. Чтобы подключить заголовочный файл в программу на C++, например iostream необходимо написать:

1. include #iostream,h;
2. #include <> с iostream.h внутри скобок
3. #include <> с iostream внутри скобок
4. include (iostreamh)

15. Тело любого цикла выполняется до тех пор, пока его условие ...

1. ложно
2. истинно
3. у цикла нет условия

16. Общий формат оператора множественного выбора - switch

1.

```
switch (switch_expression)
{
    case constant1: statement1; [break;]
    case constant2: statement2; [break;]
    case constantN: statementN; [break;]
    [default: statement N+1;]
```

}

2.

```
switch (switch_expression)
```

```
{
```

```
case constant1: statement1; [break;]
```

```
case constant2: statement2; [break;]
```

```
case constantN: statementN; [break;]
```

```
[else: statement N+1;]
```

```
}
```

3.

```
switch (switch_expression)
```

```
{
```

```
case constant1, case constant2: statement1; [break;]
```

```
case constantN: statementN; [break;]
```

```
[default: statement N+1;]
```

```
}
```

17. Какую функцию должны содержать все программы на C++?

1. start()

2. main()

3. system()

4. program()

18. Какой из ниже перечисленных операторов, не является циклом в C++?

1. repeat until

2. while

3. do while

4. for

19. Какой из следующих операторов - оператор сравнения двух переменных?

1. :=

2. =

3. equal

4. ==

20. Чему будет равна переменная a, после выполнения этого кода `int a; for(a = 0; a < 10; a++) {}`?

1. 10

2. 9

3. 1

21. Каков результат работы следующего фрагмента кода?

```
int x = 0;
```

```
switch(x)
```

```
{
```

```
case 1: cout << "Один";  
  
case 0: cout << "Нуль";  
  
case 2: cout << "Привет мир";  
  
}
```

1. Один
2. Нуль
3. НульПривет мир
4. Привет мир

22. Какой служебный знак ставится после оператора case ?

1. .
2. :
3. ;
4. -

23. Выберите правильный вариант объявления константной переменной в C++, где type - тип данных в C++ variable- имя переменной value - константное значение

1. const type variable = value;
2. const type variable := value;
3. const variable = value;

24. Какими знаками заканчивается большинство строк кода в Си++?

1. , (запятая)
2. . (точка)
3. ; (точка с запятой)
4. : (двоеточие)

25. Какой из перечисленных типов данных не является типом данных в C++?

1. double
2. real
3. int
4. float

26. Какое значение, по умолчанию, возвращает программа операционной системе в случае успешного завершения?

1. 0
2. -1
3. Программа не возвращает значение.
4. 1

27. Название C++ предложил

1. Рик Масситти

2. Кэн Томпсон
3. Бьерн Страуструп
4. Дональд Кнут

28. До каких пор будут выполняться операторы в теле цикла while ($x < 100$)?

1. Пока x строго меньше ста
2. Пока x меньше или равен ста
3. Пока x равен ста
4. Пока x больше ста

29. Укажите правильное определение функции main в соответствии со спецификацией стандарта ANSI

1. `int main(void)`
2. `int main()`
3. `void main()`
4. `void main(void)`

30. Какие преобразования типов данных не возможны без потери данных?

1. `char to float`
2. все перечисленные преобразования не возможны
3. `float to int`
4. `int to float`

31. Оператор if else позволяет определить действие ...

1. только для ложного условия
2. для истинного и ложного условий
3. только для истинного условия

32. Если условие оператора выбора ложное, то:

1. выполняется следующий оператор, сразу после оператора if
2. программа завершает работу
3. выполняется тело оператора выбора

33. В каком случае лучше всего использовать приведение типов данных?

1. при делении двух целых чисел, для того, чтобы вернуть результат с плавающей точкой
2. во всех выше указанных случаях
3. чтобы изменить тип возвращаемого значения функции
4. чтобы разрешить программе использовать только целые числа

34. Укажите операцию, приоритет выполнения которой ниже остальных.

1. `|`
2. `>>`
3. `^`
4. `<<`
5. `||`

6. ?:
7. &&
8. &

35. Укажите неправильно записанную операцию отношения

1. >=
2. =!
3. все операторы записаны правильно
4. <=

36. Какой из следующих логических операторов - логический оператор И?

1. &
2. &&
3. |
4. |&

37. Какое ключевое слово указывает, что целая переменная не может принимать отрицательные значения?

1. positive
2. нет такого зарезервированного слова
3. другое
4. unsigned
5. long

38. Какой из ниже перечисленных вариантов ответа, показывает правильно записанный оператор выбора if ?

1. if условное выражение
2. условное выражение if
3. if (условное выражение)
4. if { условное выражение }

39. Вывод данных в C++

1. cout << <переменная > , < “< строка выводится на экран>” , <выражение >
,endl;
2. cout << <переменная > << “< строка выводится на экран>” << <выражение >
<< endl;
3. cout << <переменная > , < “< строка выводится на экран>” , <выражение >
,endl;

40. В каком случае можно не использовать фигурные скобочки в операторе выбора if?

1. если в теле оператора if нет ни одного оператора
2. если в теле оператора if всего один оператор
3. если в теле оператора if два и более операторов
4. нет правильного ответа

41. Ввод данных в C++

1. `cin >> <выражение1>, <выражение2>, ...;`
2. `cin >> <выражение1> >> <выражение2> >> endl >> ...;`
3. `cin >> <выражение1> >> <выражение2> ...;`

42. Тело оператора выбора if, будет выполняться, если его условие:

1. ложно (false)
2. истинно (true)

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

1. 10
2. 11
3. 13
4. 12

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

1. порядковый номер определяется программистом
2. 18
3. 19

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

1. `cstring`
2. `ifstream.h`
3. `conio.h`
4. `cctype`

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

1. `mas;`
2. `mas[7];`
3. `mas(7);`
4. `mas[6];`

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

1. `address(a);`
2. `&a;`
3. `*a;`
4. `a ;`

48. Какая из следующих функций сравнивает две строки?

1. `stringcompare();`
2. `compare();`
3. `strcmp();`
4. `cmp();`

49. В каком из вариантов ответов объявлен двумерный массив?

1. `int anarray[20][20];`
2. `int array[20, 20];`
3. `char array[20];`
4. `array anarray[20][20];`

50. Укажите зарезервированное ключевое слово для динамического выделения памяти

1. `new`
2. `malloc`
3. `value`
4. `create`

Блок 2 (уметь)

1. В приведённом коде измените или добавьте один символ чтобы код напечатал 20 звёздочек - *.

1. `int i, N = 20;`
`for(i = 0; i < N; i--)`
`printf("*");`
2. `int i, N = 40;`
`for(i = 0; i < N; i--)`
`printf("*");`
3. `int i, N = 20;`
`for(i = 20; i < N; i--)`
`printf("*");`
4. `int i, N = 20;`
`for(i = 0; i < N; N--)`
`printf("*");`
5. `int i, N = 20;`
`for(i = 19; i < N; i--)`
`printf("*");`

2. Что будет напечатано?

```
int main()
{
    for (int i = 0; i < 4; ++i)
    {
        switch (i)
        {
            case 0 : std::cout << "0";
            case 1 : std::cout << "1"; continue;
            case 2 : std::cout << "2"; break;
            default : std::cout << "D"; break;
        }
    }
}
```

```

        std::cout << ".";
    }
    return 0;
}

```

1. Ошибка компиляции в строке 10
2. 0.1.2.
3. 01.2.D.
4. 0112.D.
5. 011.2.D

3. Что появится на экране, после выполнения этого фрагмента кода?

```

int a = 1, b = 2;
if (a == b);
cout << a << " = " << b << endl;

```

1. вывод на экран не выполнится
2. 1 = 2
3. синтаксическая ошибка
4. a = b

4. Какое значение будет напечатано

```

#include
int main(int argc, char** argv)
{
    int x = 0;
    int y = 0;

    if (x++ && y++)
    {
        y += 2;
    }

    std::cout << x + y << std::endl;

```

```

return 0;
}

```

1. 3
2. 4
3. 2
4. 1

5. Переменная x может быть доступна в другом блоке программы?

```

int main(int argc, char** argv)
{
    if ( argc > 2 )
    {
        int x = 5;

```

```

    }
    else
    {

    }

    return 0;
}
1.    да
2.    нет

```

6. Что появится на экране, после выполнения этого фрагмента кода?

```

int a = 1, b =2;
if (a == b);
cout << a << " = " << b << endl;
1.    синтаксическая ошибка
2.    вывод на экран не выполнится
3.    1 = 2
4.    a = b

```

7. Какое значение будет напечатано, в результате выполнения программы?

```

#include <iostream>

int main()
{
    int x = 3;

    switch(x)
    {
        case 0:
            int x = 1;
            std::cout << x << std::endl;
            break;
        case 3:
            std::cout << x << std::endl;
            break;
        default:
            x = 2;
            std::cout << x << std::endl;
    }

    return 0;
}

```

1. 2
2. 0
3. 3

- 4. ничего не напечатается, программа вообще не будет работать
- 5. 1

8. Какое значение будет содержать переменная x?

```
#include
```

```
int x;
```

```
int main()
```

```
{  
    int y;  
    std::cout << x << std::endl;  
    std::cout << y << std::endl;  
    return 0;  
}
```

- 1. неопределённое
- 2. 0

9. 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;  
}</iostream>
```

- 1. 21
- 2. синтаксическая ошибка
- 3. 15
- 4. 9

10. Укажите правильное объявление указателя в C++

1. `int *x;`
2. `int x;`
3. `ptr x;`
4. `int &x;`

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

1. `arr;`
2. `arr[0];`
3. `arr[1];`
4. `&arr;`

12. Укажите зарезервированное ключевое слово для высвобождения выделенной памяти

1. `clear`
2. `delete`
3. `remove`
4. `free`

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

1. `char *a; a = new char[20];`
2. `delete [] a;`
3. `delete a[];`
4. `delete a;`

14. Допустим, у нас есть код

```
char arr[8];
```

```
cin >> arr;
```

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

1. `Hello`
2. `Hello W`
3. `Hello Wo`
4. Другой ответ
5. `Hello World`

15. Укажите корректное определение строковой переменной

1. `string mystr[20];`
2. `char mystr[20];`
3. `string[20] mystr;`
4. `string mystr;`

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

1. `read(x);`
2. `getline(x, 100, 'n');`
3. `cin.getline(x, 100, 'n');`

4. `cin.getline(100, x, 'n');`

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

1. `a = '3';`
2. `a = 3;`
3. `a = "3";`

18. Укажите правильное объявление массива!

1. `int anarray[10];`
2. `array an array[10];`
3. `int anarray;`
4. `anarray{10};`

19. Правильное объявление переменной, типа структуры `foo`!

1. `foo;`
2. `int foo;`
3. `struct foo;`
4. `foo var;`

20. При определении структуры необходимо использовать следующее ключевое слово

1. `structure`
2. `struct`
3. `record`
4. `object`

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

1. `b->var;`
2. `b-var;`
3. `b>var;`
4. `b.var;`

22. 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;
```

```
}
```

1. неопределенное
2. 13
3. 12

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

1. ofstream
2. ifstream
3. другое
4. input_file

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

1. fstream
2. iostream
3. ifstream
4. infstream

25. Как организовать запись в файл?

1. a_file.out("запись");
2. a_file.printf("запись");
3. a_file="запись";
4. a_file << "запись";

26. Выберите правильное (полное) определение функции.

1.

```
int funct(int x)
{
return x = x + 1;
}
```
2.

```
void funct(x)
{
cout << "Hello"
}
```
3.

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

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

1. sizeof, const, typedef, static, voided, enum, struct, union
2. char, int, float, doubled, short, long, unsigned, signed
3. defaulted, goto, return, extern, private, public, protected
4. if, else, for, while do, switch, continue, break

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

1. int funct();;
2. funct x, y;
3. funct;
4. funct();

29. 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;
}
```

1. 3
2. 11
3. 8
4. ошибка компиляции

30. Что из нижеперечисленного не является прототипом функции?

1. int funct(char x, char y);

- 2. char x();
- 3. double funct(char x)
- 4. void funct();

31. 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;
}
```

- 1. 3
- 2. 5
- 3. ошибка компиляции
- 4. 6

32. 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;
}
```

- 1. 5
- 2. 20

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

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

1. 0
2. неопределенное
3. 5

34. Какие из следующих утверждений верны?

1. встроенные функции не должны содержать более 10 строк кода.
2. встроенные функции должны возвращать значение.
3. компилятор может проигнорировать объявление встроенной функции.
4. встроенные функции не могут возвращать значения.

35. Для чего используются встроенные функции?

1. Для удаления ненужных функций
2. Для упрощения файла с исходным кодом
3. для увеличения скорости работы программы
4. Чтобы уменьшить размер программы

36. Каков результат работы следующего фрагмента кода?

```
int x = 0;

switch(x)
{
    case 1: cout << "Один";
    case 0: cout << "Ноль";
    case 2: cout << "Привет мир";
}
```

1. Привет мир
2. Один
3. НольПривет мир
4. Ноль

37. Чему будет равна переменная a, после выполнения этого кода `int a; for(a = 0; a < 10; a++) {}`?

1. 10
2. 1
3. 9

38. В приведённом коде измените или добавьте один символ чтобы код напечатал 20 звёздочек - *.

- 1.

```

int i, N = 20;
for(i = 0; i < N; i++)
    printf("*");
2.
int i, N = 20;
for(i = 0; i < N; N--)
    printf("*");
3.
int i, N = 20;
for(i = 20; i < N; i++)
    printf("*");
4.
int i, N = 40;
for(i = 0; i < N; i++)
    printf("*");
5.
int i, N = 20;
for(i = 19; i < N; i++)
    printf("*");

```

39. Укажите блок кода, в котором переменная `y` доступна

```

1. int main(int argc, char** argv)
2. {
3.
4. if ( argc > 10 )
5. {
6. }
7. else if (int y = argc - 1 )
8. {
9.
10. }
11. else
12. {
14. }
15.
16. return 0;
17. }

```

1. строки 8 -15
2. строки 8 -17
3. строки 4 -17
4. строки 4 -15
5. строки 8 -11

40. Какое значение будет содержать переменная `x`?

```
#include
```

```
int x;
```

```
int main()
```

```
{  
    int y;  
    std::cout << x << std::endl;  
    std::cout << y << std::endl;  
    return 0;  
}
```

1. неопределённое

2. 0

41. Какое значение будет напечатано, в результате выполнения программы?

```
#include <iostream>
```

```
int main()
```

```
{  
    int x = 3;  
  
    switch(x)  
    {  
        case 0:  
            int x = 1;  
            std::cout << x << std::endl;  
            break;  
        case 3:  
            std::cout << x << std::endl;  
            break;  
        default:  
            x = 2;  
            std::cout << x << std::endl;  
    }
```

```
    return 0;  
}</iostream>
```

1. 2

2. 0

3. 3

4. ничего не напечатается, программа вообще не будет работать

5. 1

42. Что появится на экране, после выполнения этого фрагмента кода?

```
int a = 1, b =2;
```

```
if (a == b);  
cout << a << " = " << b << endl;
```

1. синтаксическая ошибка
2. вывод на экран не выполнится
3. 1 = 2
4. a = b

43. Какое значение будет напечатано?

```
#include
```

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

```
{  
    int x = 0;  
    int y = 0;  
  
    if (x++ && y++)  
    {  
        y += 2;  
    }  
  
    std::cout << x + y << std::endl;
```

```
    return 0;  
}
```

1. 3
2. 4
3. 2
4. 1

44. Результат выполнения следующего фрагмента кода: `cout << 22 / 5 * 3;`

1. другое
2. 12
3. 1
4. 1.47
5. 13.2

45. Это значение 5.9875e17 может быть сохранено в переменной, типа:

1. int
2. short
3. bool
4. long
5. float

46. Результат выполнения следующего фрагмента кода: `!((1 || 0) && 0)`

1. результат не может быть заранее определен

2. 1
3. 0

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

1. arglen
2. argv
3. count
4. argc

48. Какие из следующих функций являются встроенными?

1. inline void foo() { }
2. void foo() inline { }
3. нет правильного ответа
4. inline: void foo() { }

49. Что такое ARGV[0]?

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

50. Какой тип данных имеет переменная ARGV?

1. это не переменная
2. char **
3. int
4. char *

Максимальная сумма баллов, набираемая студентом по дисциплине «Прикладное программное обеспечение» равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	Уровень сформированности компетенций
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	Высокий уровень
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов,	Продвинутый уровень

		некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	
50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	<i>Пороговый уровень</i>
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	Компетенции не сформированы