

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Волхонов Михаил Станиславович
Должность: Ректор
Дата подписания: 23.12.2024 15:14:56
Уникальный программный ключ:
40a6db1879d6a9ee29ec8e0ffb2f95e4614a0998

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

Утверждаю:

И.о. декана электроэнергетического
факультета

Николай
Александров
ич Климов

Подписано цифровой
подписью: Николай
Александрович Климов
Дата: 2024.09.11 16:06:27
+03'00'

/Климов Н.А./

11 сентября 2024 года

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине**

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

Специальность 09.02.07 Информационные системы и программирование

Квалификация выпускника программист

Форма обучения очная

Срок освоения ППСЗ 3 года 10 месяцев

На базе основного общего образования

Фонд оценочных средств предназначен для оценивания сформированности компетенций по дисциплине «Основы алгоритмизации и программирования».

Разработчик: Алексей Сергеевич Яблоков Преподаватель кафедры информационных технологий в электроэнергетике А.С. Яблоков

Подписано цифровой подписью: Алексей Сергеевич Яблоков
Дата: 2024.09.05 14:44:55 +03'00'

Утвержден на заседании кафедры информационных технологий в электроэнергетике, протокол № 1 от 05.09.2024

Заведующий кафедрой Н.А. Климов Николай Александрович Климов

Подписано цифровой подписью: Николай Александрович Климов
Дата: 2024.09.05 14:45:08 +03'00'

Согласовано: Алексей Сергеевич Яблоков Председатель методической комиссии электроэнергетического факультета протокол № 7 от 10.09.2024

А.С. Яблоков Алексей Сергеевич Яблоков

Подписано цифровой подписью: Алексей Сергеевич Яблоков
Дата: 2024.09.10 15:05:39 +03'00'

Результаты освоения дисциплины: «Основы алгоритмизации и программирования»

ППССЗ (СПО) по специальности:
09.02.07 Информационные системы и программирование

Таблица 1

Коды компетенций по ФГОС	Компетенции	Результат освоения
1	2	3
Общие компетенции		
ОК 02.	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	<p>Знать номенклатуру информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации</p> <p>Уметь определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска</p> <p>Владеть навыками использования современных средств поиска, анализа и интерпретации информации; информационными технологиями для выполнения профессиональной деятельности</p>
ОК 09.	Пользоваться профессиональной документацией на государственном и иностранном языках	<p>Знать правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности</p> <p>Уметь понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы</p> <p>Владеть навыком использования профессиональной документации на государственном и иностранном языках</p>

1	2	3
Профессиональные компетенции		
ПК 1.1.	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования Уметь формировать алгоритмы разработки программных модулей в соответствии с техническим заданием; оценивать сложность алгоритма Владеть навыками разработки алгоритмов решения поставленной задачи и реализации его средствами автоматизированного проектирования
ПК 1.2.	Разрабатывать программные модули в соответствии с техническим заданием	Знать основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования. Уметь создавать программу по разработанному алгоритму как отдельный модуль Владеть навыком разработки кода программного продукта на основе готовой спецификации на уровне модуля
ПК 2.4.	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения	Знать модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения. Уметь использовать выбранную систему контроля версий; анализировать проектную и техническую документацию. Владеть навыками разработки тестовых сценарии программного средства
ПК 2.5.	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	Знать модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения Уметь использовать методы для получения кода с заданной функциональностью и степенью качества; анализировать проектную и техническую документацию Владеть навыками инспектирования разработанных программных модулей на предмет соответствия стандартам кодирования

Требования к результатам освоения дисциплины:

иметь навык:

Н₁ – создания программных продуктов

уметь:

У₁ – использовать основные элементы языка и принципы процедурного программирования для создания программных продуктов

знать:

З₁ – основы алгоритмизации, синтаксис языка программирования, типы данных, операторы языка

**Паспорт
фонда оценочных средств**

Таблица 2

№ п/п	Контролируемые дидактические единицы	Контролируемые компетенции (или их части)	Наименование оценочных средств		
			Тесты, кол-во заданий	Другие оценочные средства	
				вид	кол-во заданий
1	Раздел 1. Основы алгоритмизации	ОК 02 ОК 09 ПК 1.1	13	Опрос	16
2	Раздел 2. Программирование	ОК 02 ОК 09 ПК 1.1 ПК 1.2 ПК 2.4 ПК 2.5	190	Опрос	124
3	Разделы 1- 2	ОК 02 ОК 09 ПК 1.1 ПК 1.2 ПК 2.4 ПК 2.5		Контрольная работа	10
Всего:			203		150

Методика проведения контроля по проверке базовых знаний по дисциплине «Основы алгоритмизации и программирования»

Раздел 1. Основы алгоритмизации Контролируемые компетенции ОК 02; ОК 09; ПК 1.1

Вопросы для устного опроса

1. Дайте определение алгоритма. Перечислите свойства алгоритма.
2. Назовите отличия программного способа записи алгоритмов от других способов.
3. Назовите базовые алгоритмические структуры и дайте им краткую характеристику.
4. Дайте определение цикла с заданным числом повторений. Когда целесообразно применять циклы этого вида?
5. Что такое итерационные циклы? Когда возникает необходимость в их использовании?
6. Определите основные отличия между циклами с постусловием и предусловием. Как они выполняются?
7. Что называется рекуррентной формулой? Когда она применяется?
8. Дайте определение массива. Поясните, почему для хранения его элементов используется непрерывная память.
9. Можно ли при вводе или выводе элементов массива использовать цикл с предусловием или с постусловием?
10. Дайте определение двумерного массива. Поясните особенности хранения элементов двумерного массива.
11. Почему при составлении алгоритмов для решения задач с использованием двумерного массива применяется вложенный цикл?
12. Перечислите простые алгоритмы сортировки и укажите их основные отличия.
13. Почему сортировка включения является неэкономным методом?
14. Чем сортировка Шелла отличается от сортировки простыми вставками?
15. Назовите метод сортировки, который является лучшим среди простых методов. Поясните, за счет чего это достигается.
16. Почему алгоритм бинарного поиска превосходит «слепой» поиск? Какими характеристиками должен обладать массив, в котором применяется алгоритм бинарного поиска?

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который прочно усвоил программный материал в полном объеме, исчерпывающе, грамотно и логически стройно его излагает, четко формулирует основные понятия, приводит соответствующие примеры, уверенно владеет материалом.

Оценка «хорошо» выставляется обучающемуся, который твердо усвоил программный материал, грамотно и по существу излагает его без существенных ошибок, правильно применяет теоретические положения при решении конкретных задач, с небольшими погрешностями приводит формулировки определений, по ходу изложения допускает небольшие пробелы, не искажающие содержания ответа.

Оценка «удовлетворительно» выставляется обучающемуся, который не совсем твердо владеет программным материалом, знает основные теоретические положения изучаемой темы, при ответах допускает малосущественные погрешности, искажения логической последовательности при изложении материала, неточную аргументацию теоретических положений, испытывает затруднения при ответе на дополнительные вопросы.

Оценка «неудовлетворительно» выставляется обучающемуся, имеющему серьезные пробелы в знании учебного материала, допускающему принципиальные ошибки при ответе на вопросы.

Тестовые задания

Выберите один правильный вариант:

Алгоритмом можно считать:

список класса в журнале
 +описание процесса решения задачи
 рецепт блюда
 расписание уроков в школе

Свойство алгоритма, означающее, что данный алгоритм применим к решению целого класса задач – это ...

определённость
 понятность
 результативность
 +массовость

Свойство алгоритма, означающее, что он всегда приводит к результату через конечное, возможно, очень большое, число шагов – это ...

дискретность
 понятность
 +результативность
 массовость

Свойство алгоритма, означающее, что путь решения задачи разделён на отдельные шаги – это ...

+дискретность
 определённость
 результативность
 массовость

Свойство алгоритма, означающее, что путь решения задачи определён вполне однозначно, на любом шаге не допускаются никакие двусмысленности и недомолвки – это ...

дискретность
 понятность
 +определённость
 результативность

Величины, значения которых меняются в процессе исполнения алгоритма, называются ...

постоянными
 константами
 +переменными
 табличными

Величиной целого типа является:

+количество мест в зрительном зале
 рост человека
 марка автомобиля
 площадь государства

Ниже представлена схема ... алгоритма



+линейного
разветвляющегося
циклического
вспомогательного

Ниже представлена схема ... алгоритма



линейного
+разветвляющегося с полным ветвлением
циклического
разветвляющегося с неполным ветвлением

Ниже представлена схема алгоритма – ...



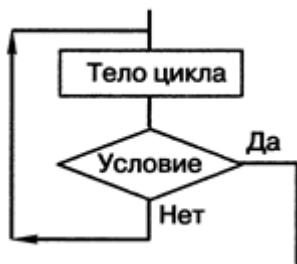
цикл с параметром
+цикл с заданным условием продолжения работы
цикл с заданным условием окончания работы
цикл с заданным числом повторений

Ниже представлена схема алгоритма – ...



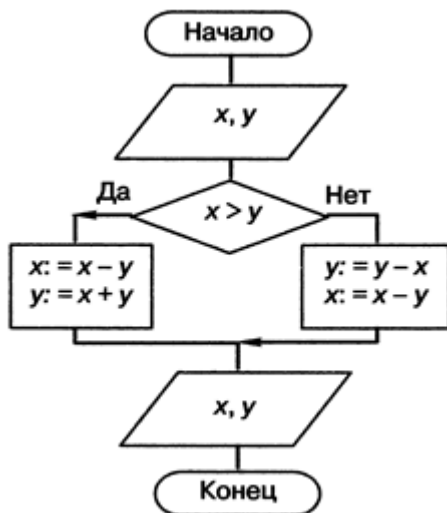
цикл с заданным условием продолжения работы
цикл с заданным условием окончания работы
цикл с постусловием
+цикл с заданным числом повторений

Ниже представлена схема алгоритма – ...



цикл с заданным условием продолжения работы
 +цикл с заданным условием окончания работы
 цикл с заданным числом повторений
 цикл с предусловием

При начальных значениях переменных $x = 10$ и $y = 15$, после выполнения алгоритма переменные x и y будут равны:



+ -5, 10
 5, 20
 10, 15
 5, 5
 -5,5

Методика проведения контроля

Параметры методики	Значение параметра
Предел длительности всего контроля	10 минут
Последовательность выбора вопросов	Случайная
Предлагаемое количество вопросов	10

Раздел 2. Программирование

Контролируемые компетенции ОК 02; ОК 09; ПК 1.1; ПК 1.2; ПК 2.4; ПК 2.5

Вопросы для устного опроса

1. Что такое язык программирования?
2. Какие парадигмы программирования поддерживает язык C++?
3. Что такое лексема?
4. Что такое идентификатор?
5. Какие существуют правила построения идентификаторов в языке C++?
6. Что такое константа?
7. Что такое переменная?
8. Как в программе на языке C++ объявляются переменные?
9. Как в коде программы выделяются комментарии?
10. Для чего используется директива include?
11. Как в языке C++ осуществить вывод и ввод информации?
12. Какие встроенные простые типы данных есть в языке C++?
13. Какие арифметические операции встроены в язык C++?
14. Что такое операторы инкремента и декремента?
15. Чем постфиксная форма операторов инкремента/декремента отличается от префиксной?
16. Какие операторы ветвления есть в языке C++?
17. Как работает оператор if?
18. Как записывается операция равенства в языке C++?
19. Как объединить в блок несколько операторов?
20. Что такое тернарный оператор, как он записывается и какой оператор может заменять?
21. Как работает оператор switch?
22. Чем оператор switch отличается от if?
23. Для чего в теле оператора switch используется оператор break?
24. Для чего используются операторы цикла?
25. Как называется один проход цикла?
26. Что такое счетчик цикла?
27. Чем цикл с предусловием отличается от цикла с постусловием?
28. Как в языке C++ выглядит синтаксис циклов с предусловием и постусловием?
29. При каком условии будет выполняться цикл while?
30. Какова структура цикла for и как он работает?
31. Какие функции выполняют операторы: break, continue, goto, return, throw?
32. Для чего в программировании используются подпрограммы?
33. Что такое функция в языке C++?
34. Какой общий вид описания функций в C++?
35. Как называются переменные, объявленные внутри функции?
36. Когда создаются и уничтожаются переменные, объявленные внутри функции?
37. Как объявляются статические переменные функции и чем они отличаются от обычных локальных переменных?
38. Что такое формальные переменные функции?
39. Чем передача параметров по значению отличается от передачи параметров по ссылке?
40. Какой вид передачи аргументов в функцию используется по умолчанию?
41. При передаче параметров по значению может ли код функции изменить аргументы, указанные при ее вызове?
42. Как объявить функцию чтобы она возвращала результат типа int?
43. Для чего в функциях используется оператор return?
44. Какой результат возвращает функция типа void?
45. Для чего используется прототип функции?
46. Следует ли использовать прототип если определение функции размещено перед ее первым вызовом?

47. Для какой функции никогда не используется прототип?
48. Что такое массив?
49. Как объявляется массив в C++?
50. Может ли размер массива задаваться переменной?
51. С какого элемента начинается индексация массивов в языке C++?
52. Как осуществить инициализацию массива при объявлении?
53. Как произвести заполнение массива с клавиатуры?
54. Как произвести поиск максимального элемента массива?
55. Что такое сортировка массива?
56. Как осуществляется сортировка массива методом прямого выбора?
57. Как осуществляется сортировка массива методом пузырька?
58. Как в C++ объявляется двумерный массив?
59. Как произвести инициализацию двумерного массива при объявлении?
60. Как заполнить двумерный массив случайными числами?
61. Как вывести двумерный массив на экран в виде таблицы?
62. Какой последний индекс будет у одномерного массива целых чисел из пяти элементов?
63. Как узнать адрес переменной в памяти?
64. Что такое указатели в C++?
65. Как выглядит объявление указателей?
66. Какой тип будет у указателя на переменную типа double?
67. Как узнать значение переменной с помощью указателя на нее?
68. Какие арифметические операции можно применять к указателям?
69. Можно ли сравнивать указатели между собой?
70. Как сделать указатель на массив?
71. На что указывает имя массива?
72. Как с помощью указателя вывести значение второго элемента массива?
73. Как использовать указатель в качестве аргумента функции? Для чего это делается?
74. Чем указатель на константу отличается от константного указателя?
75. Как выглядит объявление указателя на функцию?
76. Что такое ссылка в языке C++?
77. Как выглядит объявление ссылок?
78. Для чего используют ссылки в аргументах функции?
79. Какие два вида строк используются в языке C++?
80. Что такое строковая константа и как выглядит ее запись?
81. Как считать строку типа char* с клавиатуры вместе с пробелами?
82. Извлекает ли функция cin.get() из потока ввода символ перевода строки (\n)?
83. В каком заголовочном файле находится функция strcmp(s1,s2)? Какие функции она выполняет?
84. С помощью какого оператора осуществляется соединение строк типа string?
85. Для чего используется функция getline?
86. Какую функцию выполняет метод класса s1.c_str() string?
87. Для чего в C++ используются перечисления? Как выглядит их запись?
88. Могут ли совпадать имена элементов перечисления?
89. Что такое структура в C++? Как выглядит ее описание?
90. Что такое поле структуры?
91. Как осуществить инициализацию структуры?
92. С помощью каких операторов осуществляется доступ к полям структуры?
93. Какие функции выполняет оператор typedef?
94. Что такое инкапсуляция?
95. Что такое класс в C++? Как выглядит описание класса?
96. Что такое поля и методы класса?
97. Какие существуют спецификаторы доступа к членам класса и какие функции они выполняют?

98. Возможна ли инициализация полей класса при их описании?
99. Что такое конструктор, какие функции он выполняет и как выглядит его описание?
100. Что такое деструктор, какие функции он выполняет и как выглядит его описание?
101. Что такое конструктор копирования, какие функции он выполняет и как выглядит его описание?
102. Как создать экземпляр класса?
103. С помощью каких операторов осуществляется доступ к элементам объекта?
104. Можно ли напрямую обратиться к закрытым полям объекта из внешнего кода программы?
105. Как объявить константный метод класса?
106. На что ссылается указатель this?
107. Что такое дружественные функции и для чего они используются?
108. Как объявить статические поля класса?
109. Как объявить статический метод класса и как его потом вызвать в коде программы?
110. Какие переменные хранятся в статической памяти?
111. Какие переменные хранятся в стеке?
112. В какой области памяти разработчик может выделять место по своей инициативе в ходе работы программы?
113. С помощью каких операторов и как осуществляется выделение и освобождение динамической памяти?
114. Как выглядит выделение и освобождение памяти для массивов?
115. Какие элементы класса можно использовать для автоматического выделения и освобождения динамической памяти при создании и удалении объекта?
116. Что такое полиморфизм?
117. Что такое перегрузка функций?
118. Чем должны отличаться перегруженные функции друг от друга?
119. Как объявляются аргументы по умолчанию в функции?
120. Где, по отношению к обычным аргументам функции, должны располагаться аргументы по умолчанию?
121. Как в общем виде выглядит перегрузка операторов?
122. Чем отличается перегрузка операторов префиксного и постфиксного инкремента?
123. Может ли оператор присваивания перегружаться с помощью дружественной функции класса?
124. Можно ли с помощью перегрузки изменить приоритет операторов?

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который прочно усвоил программный материал в полном объеме, исчерпывающе, грамотно и логически стройно его излагает, четко формулирует основные понятия, приводит соответствующие примеры, уверенно владеет материалом.

Оценка «хорошо» выставляется обучающемуся, который твердо усвоил программный материал, грамотно и по существу излагает его без существенных ошибок, правильно применяет теоретические положения при решении конкретных задач, с небольшими погрешностями приводит формулировки определений, по ходу изложения допускает небольшие пробелы, не искажающие содержания ответа.

Оценка «удовлетворительно» выставляется обучающемуся, который не совсем твердо владеет программным материалом, знает основные теоретические положения изучаемой темы, при ответах допускает малосущественные погрешности, искажения логической последовательности при изложении материала, неточную аргументацию теоретических положений, испытывает затруднения при ответе на дополнительные вопросы.

Оценка «неудовлетворительно» выставляется обучающемуся, имеющему серьезные пробелы в знании учебного материала, допускающему принципиальные ошибки при ответе на вопросы.

Тестовые задания

Выберите один правильный вариант:

Минимально значимая единица текста программы

- +лексема
- пробел
- бит
- байт

В каком из вариантов идентификатор построен правильно?

- 3varN
- var*N3
- +varN3
- переменнаяN3

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 c++ - 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 C++ ...

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

В языке C++ каждый оператор заканчивается знаком

- ,
- ^
- ~
- +;

Целочисленным типом данных языка C++ является

+int
char
double
string

К вещественному типу данных в языке C++ 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...

/
+%
*
&

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 *enter*

+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
```

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

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

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 никогда 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 никогда 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

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

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);
```

+А
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);
```

+а
А
з
0

Как узнать длину строки типа string?

с помощью функции strlen
+с помощью метода lenght
с помощью метода find
с помощью метода erase

Как осуществить поиск подстроки в строке типа string?

с помощью функции strlen
с помощью метода lenght
+с помощью метода find
с помощью метода erase

Как удалить символы в строке типа string?

с помощью функции strlen
с помощью метода lenght

с помощью метода find
 +с помощью метода erase

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

сложение чисел
 +конкатенацию строк
 увеличения размера строки

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

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

```

```

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}};
```

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

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

iostream
+fstream
cmath
iomanip

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

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

Методика проведения контроля

Параметры методики	Значение параметра
Предел длительности всего контроля	10 минут
Последовательность выбора вопросов	Случайная
Предлагаемое количество вопросов	10

Билеты к экзамену по разделам 1-2

Контролируемые компетенции ОК 02; ОК 09; ПК 1.1; ПК 1.2; ПК 2.4; ПК 2.5

Вариант 1

Создайте массив из десяти целых чисел. Заполните его случайными числами и выведите на экран. Умножьте каждый элемент в массиве на 5. Выведите получившийся массив на экран.

Вариант 2

Создайте массив из десяти целых чисел. Заполните его случайными числами и выведите на экран. Найдите среднее арифметическое, и выведите на экран первый элемент массива, умноженный на среднее значение.

Вариант 3

Создайте двумерный массив 4 x 4. Заполните его случайными числами и выведите на экран. Выведите на экран максимальный элемент массива и его индексы и минимальный элемент массива с индексами.

Вариант 4

Создайте двумерный массив 4 x 4. Заполните его случайными числами и выведите на экран. Замените все элементы выше главной диагонали нулями и выведите этот массив на экран.

Вариант 5

Создайте двумерный массив 4 x 4. Заполните его случайными числами и выведите на экран. Найдите сумму главной диагонали и выведите результат на экран.

Вариант 6

Напишите программу, в которой создается строка (string). Программа просит ввести строку с клавиатуры, затем в этой строке все символы пробела меняет на символы подчеркивания. Результат выводится на экран.

Вариант 7

Напишите программу, в которой создается строка (string). Программа просит ввести строку с клавиатуры, затем удаляет первый символ в каждом слове и выводит результат на экран.

Вариант 8

Напишите программу, в которой создается строка (string). Программа просит ввести строку с клавиатуры, затем удаляет все гласные буквы и выводит результат на экран.

Вариант 9

Напишите программу, в которой создается целочисленная переменная. Ей присваивается случайное число от 0 до 100. Программа в цикле просит ввести число, если введенное число

меньше случайного, то программа выводит сообщение «задуманное число больше». Если введенное число больше, программа пишет «задуманное число меньше». Если введенное число совпадает, программа пишет «вы угадали» и завершает работу.

Вариант 10

Напишите программу, которая выводит на экран таблицу значения функции $y=x^3+3$. Начало и конец диапазона, а также шаг вводятся с клавиатуры.

Критерии оценки:

Оценка «отлично» выставляется обучающемуся, который правильно выполняет все задания, успешно применяет теоретические знания к решению практических задач с использованием алгоритма как способа автоматизации деятельности.

Оценка «хорошо» выставляется обучающемуся, который: правильно выполняет не менее 80% заданий, применяет теоретические знания к решению практических задач.

Оценка «удовлетворительно» выставляется обучающемуся, который правильно выполняет не менее 60% заданий, не совсем твердо владеет материалом, испытывает затруднения при решении достаточно сложных задач.

Оценка «неудовлетворительно» выставляется обучающемуся, который правильно выполняет менее 60% заданий, имеющему серьезные пробелы в знании учебного материала, допускающему принципиальные ошибки при выполнении предусмотренных контрольных заданий

Форма промежуточной аттестации по дисциплине *экзамен*.

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 до 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).

Дополнительные контрольные испытания

для студентов, набравших менее 50 баллов (в соответствии с Положением «О модульно-рейтинговой системе»), формируются из числа оценочных средств по темам, которые не освоены студентом.