**Карта тестовых заданий**

**Компетенция**: ОПК-7: Способен создавать программы на языках высокого и низкого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ;

**Индикатор**: ОПК-7.1: Осуществляет разработку программ на языках высокого и низкого уровня

ОПК-7.2: Осуществляет обоснованный выбор инструментария программирования и способов организации программ, использует методы и инструментальные средства программирования для решения профессиональных задач

**Дисциплина**: Языки программирования

**Описание теста:**

1. Тест состоит из 70 заданий, которые проверяют уровень освоения компетенций обучающегося. При тестировании каждому обучающемуся предлагается 30 тестовых заданий по 15 открытого и закрытого типов разных уровней сложности.

2. За правильный ответ тестового задания обучающийся получает 1 условный балл, за неправильный ответ – 0 баллов. По окончании тестирования, система автоматически определяет «заработанный итоговый балл» по тесту, согласно критериям оценки

3 Максимальная общая сумма баллов за все правильные ответы составляет – 100 баллов.

4. Тест успешно пройден, если обучающийся правильно ответил на 70% тестовых заданий (61 балл).

5. На прохождение тестирования, включая организационный момент, обучающимся отводится не более 45 минут. На каждое тестовое задание в среднем по 1,5 минуты.

6. Обучающемуся предоставляется одна попытка для прохождения компьютерного тестирования.

**Кодификатором** теста по дисциплине является раздел рабочей программы «4. Структура и содержание дисциплины (модуля)»

**Комплект тестовых заданий**

**Задания закрытого типа**

**Задания альтернативного выбора**

*Выберите* ***один*** *правильный ответ*

**Простые (1 уровень)**

1. Цикл while на Python начинается :

А) x > y while {

Б) while (x > y)

**В) while x > y**

Г) while x> y {

2. Бинарный оператор c++ записывается в виде :

**А) +**

Б) ++

В) --

Г) \*

### 3. Для того чтобы вывести символ новой строки, надо:

**А) Воспользоваться специальным манипулятором endl**

Б) При выводе строки символов перевод строки добавляется автоматически

В) Использовать метод nextLine()

Г) Закончить оператор точкой с запятой

### 4. Таблица в стандарте ASCII содержит следующее количество символов: Сколько символов (включая управляющие и непечатаемые) определяется в стандарте ASCII?

А) 24

Б) 64

**В) 128**

Г) 256

### 5. Такого типа данных не существует в С++

### А) Int

**Б) Real**

В) Float

Г) Double

**Средне – сложные (2 уровень)**

**6** Вычисления, выполняемые при вызове рекурсивной функции :

А) ленивые

**Б) циклические**

В) обратные

Г) односторонние

7 Правильный синтаксис для вывода типа переменной или объекта в Python:

**А) print(type(x)**

Б) print(typeof(x)

В) print(typeof x)

Г) print(typeOf(x)

### 8 Полиморфизм времени выполнения в C++ реализован с помощью:

### А) наследования и виртуальных функций

### Б) наследования

В) шаблонов

Г) виртуальных функций и шаблонов

### 9 Бинарный оператор это :

### А) Оператор, который действует над единственным аргументом

Б) Оператор, который действует над любым количеством аргументов

В) Оператор, который действует над тремя аргументами

**Г) Оператор, который действует над двумя аргументами**

### 10 Метод для удаления любых пробелов как в начале, так и в конце строки в Python:

А) ptrim()

Б) trim()

**В) strip()**

Г) len()

### 11. Значения по умолчанию аргументов функции указываются при:

А) вызове функции

**Б) объявлении функции**

В) выходе из области видимости

Г) определении функции

### 12. Отличие delete и delete[] :

А) delete удаляет нормальный объект, а delete[] - указатель

Б) delete - это ключевое слово, а delete[] - это идентификатор

**В) delete используется для удаления единичного объекта, а delete[] удаляет массив объектов**

Г) delete - синтаксически правильно, а delete[] - ошибочная конструкция

### 13. Правильным примером исключения является:

А) Переменная не объявлена

**Б) Деление на ноль**

В) Пропущена точка с запятой

Г) Неправильное выражение

### 14. Заголовочный файл используемый при обработке исключений

**А) <exception>**

Б) <cstdlib>

В) <string>

Г) <handler>

### 15. Оператор для остановки цикла:

А) return

**Б) break**

В) exit

Г) stop

### 16. Обязательная часть объявления функции

А) Возвращаемый тип, имя функции, список аргументов

**Б) Возвращаемый тип, имя функции**

В) Список аргументов, имя функции

Г) Список аргументов, список переменных

### 17. Область видимости переменной, определенной в функции в языке С++

А) Вся программа

Б) Функция main()

**В) Только внутри блока { } функции**

Г) В файлах заголовков

### 18. Для указания переменной в С++ необходимо записать

А) var str = "Hi";

Б) float x = 32,14;

**В) char sym = 'a';**

Г) done = true;

19. Стандартная библиотека iostream в C++ подключается :

А) #include “iostream.h”

Б) #include <iostream.h>

В) #include iostream

**Г) #include <iostream>**

20. Выражение не содержащие синтаксических ошибок:

А) sin(abs(0.6(e3))

**Б) ((cos(3a+1.abs(x))))**

В) aexp(t)(2t)

Г) cos(exp,3a)\*abs(x)

21. Фрагмент кода , который выводит все элементы массива при объявлениеи int a[15],

А) for (int k = 0; k <= 15; k++) cout << a[k];

Б) for (int k = 1; k < 16; k++) cout << a[k];

**В) for (int k = 0; k < 15; k++) cout << a[k];**

Г) for (int k = 1; k <= 16; k++) cout << a[k];

### 22. Правильное утверждение относительно iostream :

**А) iostream - стандартный заголовок, а iostream.h - нестандартный**

Б) iostream - нестандартный заголовок, и iostream.h - нестандартный

В) iostream - стандартный заголовок, и iostream.h - стандартный

Г) iostream - нестандартный заголовок

**Сложные (3 уровень)**

### 23. setprecision() требует следующий заголовочный файл

**А) iomanip**

Б) cstdlib

В) conio.h

Г) console.h

### 24. Определение пользовательского типа данных как объединения (union) позволяет:

**А) Обращаться к одной и той же области памяти как к переменной разных типов**

Б) Это позволяет обращаться к различным областям памяти как к переменным одного и того же типа

В) Это позволяет обращаться к различным областям памяти как к переменным различных типов

Г) Это позволяет обращаться к одной и той же области памяти как к переменной одного и того же типа

25. Оператор === ( три знака “равно” ) :

А) Проверяет правильность введённых данных

Б) Сравнивает три переменные

В) Нет такого оператора

**Г) Сравнивает без приведения типа**

**Задания на установление соответствия**

*Установите соответствие.*

**Простые (1 уровень)**

26. Установите соответствие:

|  |  |
| --- | --- |
| 1 for | А) Выполняет указанный блок кода, пока заданное условие истинно |
| 2 while | Б) Прерывает выполнение цикла и переходит к следующей строке кода за его телом |
| 3 break | В) Используется для перебора элементов в итерируемом объекте |
|  | Г) Прерывает выполнение текущей итерации цикла и переходит к следующей |

27. Установите соответствие:

|  |  |
| --- | --- |
| 1 private | А) Члены класса, доступные только внутри класса и его наследников |
| 2 public | Б) Модификатор доступа не используется в C++. |
| 3 protected | В) Члены класса, доступные только внутри класса |
|  | Г) Члены класса, доступные из любой части программы |

**Средне-сложные (2 уровень)**

28. Установите соответствие:

|  |  |
| --- | --- |
| 1 Инкапсуляция | А) Механизм, позволяющий классам наследовать свойства и методы других классов |
| 2 Наследование | Б) Скрытие деталей реализации от других частей программы |
| 3 Полиморфизм | В) Разделение программы на независимые компоненты. |
|  | Г) Возможность объектов с одинаковой сущностью иметь различное поведение в зависимости от контекста. |

29. Установите соответствие:

|  |  |
| --- | --- |
| 1 Class | А) Шаблон для создания объектов, содержащий свойства и методы |
| 2 Object | Б) Функция, которая вызывается для доступа к свойствам и методам объекта класса |
| 3 Constructor | В) Изменяемый объект, содержащий уникальные и неупорядоченные элементы |
|  | Г) Функция, которая вызывается при создании объекта класса |

30. Установите соответствие между типами данных и их размерами в байтах в языке программирования C++:

|  |  |
| --- | --- |
| 1 bool | А) Занимает 8 байт |
| 2 char | Б) Занимает 1 байт |
| 3 float | В) Занимает 4 байта |
|  | Г) Занимает 2 байта |

31. Установите соответствие между понятиями и их описанием в языке программирования Python:

|  |  |
| --- | --- |
| 1 Lambda функция | А) Функция, которая может принимать любое количество аргументов |
| 2 Args | Б) Изменяемый объект, содержащий уникальные и неупорядоченные элементы |
| 3 Kwargs | В) Функция, которая содержит одно выражение и не имеет имени |
|  | Г) Аргументы функции, которые могут принимать любое количество значений |

32. Установите соответствие:

|  |  |
| --- | --- |
| 1 Шаблон функции | А) Объект, который не имеет конкретного типа данных, позволяющий сохранять значения различных типов данных |
| 2 Шаблон класса | Б) Функция, которая может принимать аргументы различных типов данных без использования перегрузки функций |
| 3 Указатель void | В) Указатель, который может хранить адрес объекта любого типа данных |
|  | Г) Конструкция языка C++, позволяющая создавать обобщенные функции |

33. Установите соответствие:

|  |  |
| --- | --- |
| 1 Ссылка на объект | А) Объект, который может хранить значение только 0 или 1 |
| 2 Указатель на объект | Б) Часть структуры, которая занимает заданное число бит и может использоваться для экономии памяти |
| 3 Битовое поле | В) Объект, который содержит адрес ячейки памяти |
|  | Г) Переменная, которая содержит ссылку на объект и обеспечивает доступ к этому объекту без использования его имени |

34. Установите соответствие:

|  |  |
| --- | --- |
| 1 split() | А) Возвращает новую строку, полученную путем объединения строк из последовательности |
| 2 strip() | Б) Заменяет все вхождения указанной подстроки на другую подстроку |
| 3 join() | В) Возвращает строку с пропущенными начальными и конечными символами |
|  | Г) Разбивает строку на список подстрок по разделителю и возвращает список |

**Сложные (3 уровень)**

35. Установите соответствие между левым и правым столбцами:

|  |  |
| --- | --- |
| 1 Множественное наследование в Python | А) Механизм, который позволяет наследовать методы и атрибуты от нескольких классов |
| 2 Композиция объектов в Python | Б) Механизм, который позволяет объединить несколько классов в один |
| 3 Абстрактные классы в Python | В) Класс, который не может быть создан напрямую, а используется только для наследования |
|  | Г) Механизм, который позволяет создавать новый объект путем комбинирования нескольких объектов |

**Задания открытого типа**

**Задания на дополнение**

*Напишите пропущенное слово.*

**Простые (1 уровень)**

36. Для выбора одного из нескольких вариантов действий в языке программирования С++ используется оператор …

**switch**

37. В языке программирования Python для вывода текста на экран используется функция …

**print()**

38. Для сравнения двух значений в Python необходимо использовать оператор …

==

39. В С++ для хранения символов необходимо использовать тип данных …

**char**

40. Для получения длины строки в Python используется функция …

**len()**

41. Для объявления указателя в С++ используется оператор …

**\***

42. для выполнения операции присваивания в Python используется оператор …

**=**

**Средне-сложные (2 уровень)**

43. В Python для работы с динамической памятью используется функция

i**del()**

44. В С++ для ввода данных с клавиатуры используется функция …

**сin**

45. В С++ для работы с файлами используется класс…

**ifstream**

46. В С++ для объявления константы используется ключевое слово …

**const**

47. В С++ для работы со строками используется следующий тип данных …

**std::string**

48. В Python для динамического выделения памяти используется метод …

**new()**

49. В С++ для получения размера массива используется функция …

**sizeof()**

50. для возврата значения из функции в Python используется оператор …

**return**

51. Для работы с динамическими массивами в C++ используется класс …

**vector**

52. В С++ для перегрузки оператора вывода в поток используется следующий оператор …

<<

53. В языке Python для работы с датами и временем используется модуль …

**datetime**

54. В языке С++ для работы с датами и временем используется класс …

**std::chrono**

55. В С++ для использования функций из другого файла необходимо использовать директиву …

**#include**

56. Для открытия файла в режиме записи в С++ используется режим …

**ios::out**

57. В Python для открытия файла в режиме чтения используется режим …

**'r'**

58. Для объединения строк в C++ используется функция …

**strcat**

59. В Python для преобразования строки в число используется функция …

**int**

60. В С++ для перехода к следующей итерации цикла используется ключевое слово …

**continue**

61. В Python для ввода данных из консоли используется функция …

**input**

62. В C++ для замены подстроки в строке используется метод…

**replace()**

63. В Python для удаления элемента из списка используется метод …

**remove()**

64. В Python для получения минимального элемента списка используется функция …

**min()**

65. В С++ для выполнения оператора условия используется ключевое слово …

**if**

**66. В Python для использования функций из другого файла необходимо использовать ключевое слово …**

**import**

**Сложные (3 уровень)**

67. В языке Python для создания сопрограммы используется ключевое слово …

**async def**

68. В языке Python для работы с параллельными процессами используется модуль …

**multiprocessing**

69. В языке С++ для работы с файлами используется класс …

**fstream**

70. В языке С++ для работы с потоками используется класс …

**thread**

**Карта учета тестовых заданий**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Компетенция | ПК-1. Способен разрабатывать прототип информационной системы на базе типовой информационной системы. | | | |
| Индикатор | ПК-1.3 Разрабатывает web-приложения для информационных систем. | | | |
| Дисциплина | WEB-программирование. | | | |
| Уровень освоения | Тестовые задания | | | Итого |
| Закрытого типа | | Открытого типа |
| Альтернативный выбор | Установление соответствия/ последовательности | На дополнение |
| 1.1.1 (20%) | 5 | 2 | 7 | 14 |
| 1.1.2 (70%) | 17 | 7 | 24 | 48 |
| 1.1.3 (10%) | 3 | 1 | 4 | 8 |
| Итого: | 25 шт. | 10 шт. | 35 шт. | 70 шт. |

**Критерии оценивания**

**Критерии оценивания тестовых заданий**

Критерии оценивания: правильное выполнение одного тестового задания оценивается 1 условным баллом, неправильное – 0 баллов.

Максимальная общая сумма баллов за все правильные ответы составляет наивысший балл – 100 баллов.

**Шкала оценивания результатов компьютерного тестирования обучающихся** (рекомендуемая)

|  |  |  |
| --- | --- | --- |
| Оценка | Процент верных ответов | Баллы |
| «удовлетворительно» | 70-79% | 61-75 баллов |
| «хорошо» | 80-90% | 76-90 баллов |
| «отлично» | 91-100% | 91-100 баллов |

**Ключи ответов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ тестовых заданий** | **Номер и вариант правильного ответа** |  |  | **36** | **switch** |
| **1** | **В** |  |  | **37** | **print()** |
| **2** | **А** |  |  | **38** | **==** |
| **3** | **А** |  |  | **39** | **char** |
| **4** | **В** |  |  | **40** | **len()** |
| **5** | **Б** |  |  | **41** | **звёздочка, \*** |
| **6** | **Б** |  |  | **42** | = |
| **7** | **А** |  |  | **43** | i**del()** |
| **8** | **А** |  |  | **44** | **сin** |
| **9** | **Г** |  |  | **45** | **Ifstream** |
| **10** | **В** |  |  | **46** | **const** |
| **11** | **Б** |  |  | **47** | **std::string** |
| **12** | **В** |  |  | **48** | **new()** |
| **13** | **Б** |  |  | **49** | **sizeof()** |
| **14** | **А** |  |  | **50** | **return** |
| **15** | **Б** |  |  | **51** | **vector** |
| **16** | **Б** |  |  | **52** | **<<** |
| **17** | **В** |  |  | **53** | **datetime** |
| **18** | **В** |  |  | **54** | **std::chrono** |
| **19** | **Г** |  |  | **55** | **#include** |
| **20** | **Б** |  |  | **56** | **ios::out** |
| **21** | **В** |  |  | **57** | **'r'** |
| **22** | **А** |  |  | **58** | **strcat** |
| **23** | **А** |  |  | **59** | **int** |
| **24** | **А** |  |  | **60** | **continue** |
| **25** | **Г** |  |  | **61** | **input** |
| **26** | 1- Используется для перебора элементов в итерируемом объекте  2- Выполняет указанный блок кода, пока заданное условие истинно  3- Прерывает выполнение текущей итерации цикла и переходит к следующей |  |  | **62** | **replace()** |
| **27** | 1- Члены класса, доступные только внутри класса и его наследников  2- Члены класса, доступные из любой части программы  3- Члены класса, доступные только внутри класса |  |  | **63** | **remove()** |
| **28** | 1- Скрытие деталей реализации от других частей программы  2- Механизм, позволяющий классам наследовать свойства и методы других классов  3- Возможность объектов с одинаковой сущностью иметь различное поведение в зависимости от контекста. |  |  | **64** | **min()** |
| **29** | 1- Шаблон для создания объектов, содержащий свойства и методы  2- Изменяемый объект, содержащий уникальные и неупорядоченные элементы  3- Функция, которая вызывается при создании объекта класса |  |  | **65** | **if** |
| **30** | 1- Занимает 1 байт  2- Занимает 2 байта  3- Занимает 4 байта |  |  | **66** | **import** |
| **31** | 1- Функция, которая содержит одно выражение и не имеет имени  2- Аргументы функции, которые могут принимать любое количество значений  3- Функция, которая может принимать любое количество аргументов |  |  | **67** | **async def** |
| **32** | 1- Конструкция языка C++, позволяющая создавать обобщенные функции  2- Указатель, который может хранить адрес объекта любого типа данных  3- Объект, который не имеет конкретного типа данных, позволяющий сохранять значения различных типов данных |  |  | **68** | **multiprocessing** |
| **33** | 1- Переменная, которая содержит ссылку на объект и обеспечивает доступ к этому объекту без использования его имени  2- Объект, который содержит адрес ячейки памяти  3- Часть структуры, которая занимает заданное число бит и может использоваться для экономии памяти |  |  | **69** | **fstream** |
| **34** | 1- Разбивает строку на список подстрок по разделителю и возвращает список  2- Возвращает строку с пропущенными начальными и конечными символами  3- Возвращает новую строку, полученную путем объединения строк из последовательности |  |  | **70** | **thread** |
| **35** | 1- Механизм, который позволяет наследовать методы и атрибуты от нескольких классов  2- Механизм, который позволяет создавать новый объект путем комбинирования нескольких объектов  3- Класс, который не может быть создан напрямую, а используется только для наследования |  |  |  |  |