**Ответы на вопросы испытания первого этапа Online олимпиада для участников проекта "Код будущего" по программе "Разработчик игр на Python: играй в свою игру"**

1. Найдите логическую ошибку в программе: ****
2. **в программе нету ошибок**
3. **ошибка в строке 1**
4. **ошибка в строке 2**
5. **ошибка в строке 3**
6. **ошибка в строке 5**
7. Какой тип данных в python не поддерживает оператор «+»?
8. **str**
9. **list**
10. **tuple**
11. **set**
12. Что делает оператор **in** при использовании со списками?
13. **Проверяет наличие элемента в списке**
14. Возвращает сумму всех элементов в списке
15. Удаляет элемент из списка
16. Добавляет элемент в список
17. Что является списками?
18. Последовательно сохраненные переменные
19. **Упорядоченные, неограниченные по размеру коллекции объектов произвольных типов**
20. Упорядоченные, неограниченные по размеру коллекции объектов одного типа данных
21. Не упорядоченные, неограниченные по размеру коллекции объектов произвольных типов
22. Какой результат выдаст программа в случае выполнения кода на экране?

****

1. **True**
2. False
3. Выдаст оштибку
4. Какой результат будет у выражения **3 \* "Hello"**?
5. **«HelloHelloHello»**
6. 15
7. «3\*Hello»
8. ошибка
9. Как получить длину списка в Python?
10. size(list)
11. count(list)
12. **len(list)**
13. sizeOf(list)
14. Какие из перечисленных типов данных являются изменяемыми?
15. int, str
16. **list, set**
17. tuple, bool
18. int, tuple
19. Как представлены данные в языке программирования Python?
20. В виде бинарных строк (строк байтов)
21. **В виде объектов**
22. В виде функций
23. В виде массивов
24. **Для чего используется – is?**
25. **возвращает false если переменные указывают на один и тот же объект памяти**
26. **возвращает true если переменные указывают на один и тот же объект памяти**
27. **возвращает None если переменные указывают на разные типы данных**
28. **означает что оператор is не будет выполнен при равенстве значений**Какой результат выдаст программа в случае выполнения кода на экране?

 ****

1. **10**
2. **30**
3. **Ошибка**
4. **Верно, сопоставьте определения операторов
continue -** Прерывает текущую итерацию цикла и переходит к следующей.
**break -**Прерывает выполнение цикла и выходит из него.

**return -** Возвращается из функции, передавая значение вызывающему коду.

**pass -** Ничего не делает. Этот оператор используется, когда синтаксически требуется наличие оператора, но никаких действий не требуется. Например, при создании пустого блока кода, функции или класса.

1. **Как, верно, произвести поиск слова «солнца» в переменной my\_list?**

****

1. **Никак**
2. **my\_list.find(“солнца”)**
3. **my\_list.find\_all(“солнца”)**
4. **my\_list[1].find(“солнца”)**
5. **Какой результат будет в результате выполнения данного кода?**

****

1. **None**
2. **[0, 5, 2, 5, 5, 2]**
3. **[0, 5, 2, 5]**
4. **[0, 5, 2, 3]**
5. Какой результат выдаст программа в случае выполнения кода на экране?



1. «Деление на ноль невозможно»
2. **-8**
3. -12
4. 16
5. Существует кортеж координат - coordinates = ((0, 2), (5, 44), (5, 5)), как удалить из него третью пару координат?
6. coordinates.remove(2)
7. coordinates.pop(2)
8. **никак**
9. del coordinates[2]
10. Требуется сопоставить способы импорта



import func - Блок А

from func import \* - Блок В

from functions import mutli, logic - Блок С

1. На строках 5-8 написано логическое выражение, каким образом его можно верно сократить?

****

1. **используя пример на 10 строке**
2. используя пример на 11 строке
3. используя пример на 12 строке
4. используя пример на 13 строке
5. **Данный код создает матрицу с заданным количеством строк и столбцов, определи сумму чисел в матрице:**

****

1. **8**
2. **16**
3. **32**
4. **0**
5. Какой результат выполнения данного кода?

****

1. Ошибка
2. **8**
3. 20
4. 12
5. Верно, сопоставь определения:
**zip** - объединяет элементы из нескольких итерируемых объектов в кортежи. Возвращает итератор кортежей.
**map** - применяет указанную функцию ко всем элементам указанного итерируемого объекта (например, списка) и возвращает итератор, содержащий результаты.
**isinstance** - проверяет, является ли объект экземпляром указанного класса или классов. Возвращает **True**, если объект принадлежит к указанным классам, и **False** в противном случае.
**lambda** - используется для создания анонимных (безымянных) функций. Они могут содержать только одно выражение и обычно используются вместе с функциями, которые принимают функции в качестве аргументов.

**enumerate** -используется для пронумерования элементов итерируемого объекта, возвращая кортежи вида (индекс, элемент).