

Компонент ОПОП 09.04.01 Информатика и вычислительная техника
Направленность (профиль) Компьютерный анализ и интерпретация данных,
Data Science.

Б1.О.11
ШПФР дисциплины

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Дисциплины (модуля) Прикладные библиотеки Python в науках о данных

Разработчик:
Золотов Олег Владимирович
ФИО
доцент
должность
канд. физ.-мат. наук
ученая степень,
звание

Утверждено на заседании кафедры
цифровых технологий, математики и экономики
протокол №13 от 29.06.2022г.
И.о. заведующего кафедрой ЦТМиЭ


подпись Мотина Т.Н.
ФИО

Мурманск

2022

1. Критерии и средства оценивания компетенций и индикаторов их достижения, формируемых дисциплиной (модулем)

Код и наименование компетенции	Код и наименование индикатора(ов) достижения компетенции	Результаты обучения по дисциплине (модулю)			Оценочные средства текущего контроля	Оценочные средства промежуточной аттестации
		<i>Знать</i>	<i>Уметь</i>	<i>Владеть</i>		
ОПК-2. Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	ИД-1 оПК-2 Понимает принципы и применяет методы разработки алгоритмов и программных средств	принципы и методы разработки алгоритмов и программных средств, современные информационно-коммуникационные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач в области наук о данных с использованием прикладных библиотек Python; общие принципы построения и функционирования информационных и автоматизированных систем, применяемых при решении задач в области наук о данных с использованием прикладных библиотек Python;	осуществлять обоснованный выбор современных информационно-коммуникационных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач в области наук о данных с использованием прикладных библиотек Python; разрабатывать и модернизировать программное обеспечение и элементы информационных и автоматизированных систем, применяемых при решении задач в области наук о данных с использованием прикладных библиотек Python; выполнять анализ подходов к	методами разработки алгоритмов и программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач в области наук о данных с использованием прикладных библиотек Python; навыком разработки и модернизации элементов (компонентов) программного обеспечения и информационных и автоматизированных систем, применяемых при решении задач в области наук о данных с	- комплект заданий для выполнения лабораторных работ; - типовые задания по вариантам для выполнения расчетно-графической работы; - контроль посещаемости занятий.	Зачет Результаты текущего контроля
	ИД-2 оПК-2 Использует современные интеллектуальные технологии при разработке алгоритмов и программных средств	ИД-3 оПК-2 Решает профессиональные задачи путем разработки оригинальных алгоритмов и программных средств	ИД-1 оПК-5 Понимает принципы построения и функционирования информационных и автоматизированных систем, применяемых при решении задач в области наук о данных с использованием прикладных библиотек Python;			

информационных и автоматизированных систем	систем	при решении задач в области наук о данных с использованием прикладных библиотек Python;	разработке компонент программно-аппаратных комплексов обработки информации и автоматизированного проектирования с учетом принципов их работы, использовать инструментальные средства разработки при решении задач в области наук о данных с использованием прикладных библиотек Python	использованием прикладных библиотек Python; навыком применения приемов анализа подходов к разработке компонент программно-аппаратных комплексов обработки информации и автоматизированного проектирования с учетом принципов их работы, использования инструментальных средств разработки при решении задач в области наук о данных с использованием прикладных библиотек Python		
ОПК-6. Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования	<p>ИД-1_{оПК-6} Понимает принципы работы программно-аппаратных комплексов обработки информации и автоматизированного проектирования</p> <p>ИД-2_{оПК-6} Анализирует существующие подходы к разработке компонент программно-аппаратных комплексов обработки информации и автоматизированного проектирования</p> <p>ИД-3_{оПК-6} Использует инструментальные средства разработки компонент программно-аппаратных комплексов</p>	<p>общие принципы работы программно-аппаратных комплексов обработки информации и автоматизированного проектирования, анализа подходов к разработке их компонент, инструментальные средства разработки, применяемые при решении задач в области наук о данных с использованием прикладных библиотек Python</p>				

2. Оценка уровня сформированности компетенций (индикаторов их достижения)

Показатели оценивания компетенций (индикаторов их достижения)	Шкала и критерии оценки уровня сформированности компетенций (индикаторов их достижения)			
	Ниже порогового («неудовлетворительно»)	Пороговый («удовлетворительно»)	Продвинутый («хорошо»)	Высокий («отлично»)
Полнота знаний	Уровень знаний ниже минимальных требований. Имели место грубые ошибки.	Минимально допустимый уровень знаний. Допущены не грубые ошибки.	Уровень знаний в объёме, соответствующем программе подготовки. Допущены некоторые погрешности.	Уровень знаний в объёме, соответствующем программе подготовки.
Наличие умений	При выполнении стандартных заданий не продемонстрированы основные умения. Имели место грубые ошибки.	Продемонстрированы основные умения. Выполнены типовые задания с не грубыми ошибками. Выполнены все задания, но не в полном объёме (отсутствуют пояснения, неполные выводы)	Продемонстрированы все основные умения. Выполнены все основные задания с некоторыми погрешностями. Выполнены все задания в полном объёме, но некоторые с недочётами.	Продемонстрированы все основные умения. Выполнены все основные и дополнительные задания без ошибок и погрешностей. Задания выполнены в полном объёме без недочётов.
Наличие навыков (владение опытом)	При выполнении стандартных заданий не продемонстрированы базовые навыки. Имели место грубые ошибки.	Имеется минимальный набор навыков для выполнения стандартных заданий с некоторыми недочётами.	Продемонстрированы базовые навыки при выполнении стандартных заданий с некоторыми недочётами.	Продемонстрированы все основные умения. Выполнены все основные и дополнительные задания без ошибок и погрешностей. Продемонстрирован творческий подход к решению нестандартных задач.
Характеристика сформированности компетенции	Компетенции фактически не сформированы. Имеющихся знаний, умений, навыков недостаточно для решения практических (профессиональных) задач. ИЛИ Зачетное количество баллов не набрано согласно установленному диапазону	Сформированность компетенций соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практических (профессиональных) задач. ИЛИ Набрано зачетное количество баллов согласно установленному диапазону	Сформированность компетенций в целом соответствует требованиям. Имеющихся знаний, умений, навыков достаточно для решения стандартных профессиональных задач. ИЛИ Набрано зачетное количество баллов согласно установленному диапазону	Сформированность компетенций полностью соответствует требованиям. Имеющихся знаний, умений, навыков в полной мере достаточно для решения сложных, в том числе нестандартных, профессиональных задач. ИЛИ Набрано зачетное количество баллов согласно установленному диапазону

3. Критерии и шкала оценивания заданий текущего контроля

3.1 Критерии и шкала оценивания лабораторных работ

Перечень лабораторных работ, описание порядка выполнения и защиты работы, требования к результатам работы, структуре и содержанию отчета и т.п. представлены в методических материалах по освоению дисциплины (модуля) и в электронном курсе в ЭИОС МГТУ.

Оценка/баллы	Критерии оценивания
<i>Отлично / 3</i>	Задание выполнено полностью и правильно. Отчет по лабораторной/практической работе подготовлен качественно в соответствии с требованиями. Полнота ответов на вопросы преподавателя при защите работы.
<i>Хорошо / 2</i>	Задание выполнено полностью, но нет достаточного обоснования или при верном решении допущена незначительная ошибка, не влияющая на правильную последовательность рассуждений. Все требования, предъявляемые к работе, выполнены.
<i>Удовлетворительно / 1</i>	Задания выполнены частично с ошибками. Демонстрирует средний уровень выполнения задания на лабораторную/практическую работу. Большинство требований, предъявляемых к заданию, выполнены.
<i>Неудовлетворительно / 0</i>	Задание выполнено со значительным количеством ошибок на низком уровне. Многие требования, предъявляемые к заданию, не выполнены. ИЛИ Задание не выполнено.

3.2 Критерии и шкала оценивания расчетно-графической работы

Перечень тем расчетно-графических работ, рекомендации по выполнению представлены в методических материалах по освоению дисциплины (модуля) и в электронном курсе в ЭИОС МГТУ.

В ФОС включен типовой вариант расчетно-графической работы.

С использованием рассматриваемых в программе дисциплины прикладных библиотек Python (или их альтернатив; при использовании альтернатив выполнить обоснование целесообразности их использования):

- 1) организовать получение данных модели FIRI-2018 из сети Internet (ссылку на источник данных предоставляет преподаватель);
- 2) осуществить разбор *xlsx*-файлов данных модели FIRI-2018 и привести эти данные к виду, пригодному для анализа в Python;
- 3) выполнить анализ данных (в т.ч. определение наличия / отсутствия пропущенных значений) и, в случае необходимости, преобразование данных (в том числе заполнение пропущенных данных, если потребуется);
- 4) результаты анализа сопроводить необходимым для понимания характера и особенностей данных графическим материалом.

Оценка/баллы	Критерии оценивания
<i>Отлично / 54</i>	Работа выполнена полностью, без ошибок (возможна одна неточность, описка, не являющаяся следствием непонимания материала).
<i>Хорошо / 50</i>	Работа выполнена полностью, но обоснования шагов решения недостаточны, допущена одна негрубая ошибка или два-три недочета, не влияющих на правильную последовательность рассуждений.
<i>Удовлетворительно /</i>	В работе допущено более одной грубой ошибки или более двух-трех

43	недочетов, но обучающийся владеет обязательными умениями по проверяемой теме.
Неудовлетворительно / 0	В работе есть грубые ошибки и недочеты ИЛИ Контрольная работа не выполнена.

Критерии и шкала оценивания посещаемости занятий

Посещение занятий обучающимися определяется в процентном соотношении

Баллы	Критерии оценки
10	посещаемость 75 - 100 %
5	посещаемость 50 - 74 %
0	посещаемость менее 50 %

4. Критерии и шкала оценивания результатов обучения по дисциплине (модулю) при проведении промежуточной аттестации

Критерии и шкала оценивания результатов освоения дисциплины (модуля) с зачетом

Если обучающийся набрал зачетное количество баллов согласно установленному диапазону по дисциплине (модулю), то он считается аттестованным.

Оценка	Баллы	Критерии оценивания
<i>Зачтено</i>	60 - 100	Набрано зачетное количество баллов согласно установленному диапазону
<i>Незачтено</i>	менее 60	Зачетное количество согласно установленному диапазону баллов не набрано

5. Задания диагностической работы для оценки результатов обучения по дисциплине (модулю) в рамках внутренней и внешней независимой оценки качества образования

ФОС содержит задания для оценивания знаний, умений и навыков, демонстрирующих уровень сформированности компетенций и индикаторов их достижения в процессе освоения дисциплины (модуля).

Комплект заданий разработан таким образом, чтобы осуществить процедуру оценки каждой компетенции, формируемых дисциплиной (модулем), у обучающегося в письменной форме.

Содержание комплекта заданий включает: тестовые задания.

Комплект заданий диагностической работы

ОПК-2 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач

1	<p><i>Следующий код</i></p> <pre>import itertools as it counter = it.count(2022) lst = list(counter)</pre> <p>А. создаст список, содержащий 2022 элемента Б. создаст список, содержащий один элемент, равный 2022 В. не будет завершен (произойдет ошибка по исчерпанию выделенных ресурсов) Г. нет правильного ответа</p> <p>Ответ: В. не будет завершен (произойдет ошибка по исчерпанию выделенных ресурсов)</p>
2	<p><i>В результате выполнения следующего кода</i></p> <pre>import itertools as it val = it.cycle([2022]) a = next(val) b = next(val) print(a, b)</pre> <p><i>в стандартный поток вывода будет выведено:</i></p> <p>А. 2022 2022 Б. 2022 2023 В. 2023 2024 Г. нет правильного ответа</p> <p>Ответ: А. 2022 2022</p>
3	<p><i>В результате выполнения следующего кода</i></p> <pre>import itertools as it res = it.accumulate([1,2,3]) print(list(res))</pre> <p><i>в стандартный поток вывода будет выведено:</i></p> <p>А. 6 Б. [6] В. [1, 3, 6] Г. нет правильного ответа</p>

	<p>Ответ: В. [1, 3, 6]</p>
4	<p><i>В результате выполнения следующего кода</i></p> <pre data-bbox="309 338 983 685">import itertools as it val = it.repeat(2022, times=2) a = next(val) b = next(val) c = next(val) print(a, b, c)</pre> <p><i>будет:</i></p> <p>А. напечатано 2022 2022 2022 Б. напечатано 2022 2023 2024 В. напечатано 2022 2022 None Г. порождено исключение StopIteration</p> <p>Ответ: Г. порождено исключение StopIteration</p>
5	<p><i>Для следующего кода</i></p> <pre data-bbox="309 1070 1062 1261">import itertools as it val_r = it.repeat(2022) val_c = it.cycle([2022])</pre> <p><i>выберите верное утверждение</i></p> <p>А. val_r и val_c обе являются бесконечными последовательностями (генераторами), возвращающими число 2022 Б. val_c генерирует бесконечную возрастающую последовательность, начиная с числа 2022, а val_r возвращает бесконечную последовательность, возвращающую число 2022 В. val_r генерирует бесконечную возрастающую последовательность, начиная с числа 2022, а val_c возвращает бесконечную последовательность, возвращающую число 2022 Г. нет правильного ответа</p> <p>Ответ: А. val_r и val_c обе являются бесконечными последовательностями (генераторами), возвращающими число 2022</p>
6	<p>Выберите верное утверждение:</p> <p>А. itertools.accumulate может осуществлять только суммирование Б. itertools.accumulate позволяет задавать применяемый оператор В. itertools.accumulate работает только с числовыми типами Г. itertools.accumulate работает только со списками</p>

Ответ:
Б. `itertools.accumulate` позволяет задавать применяемый оператор

7

В результате выполнения следующего кода

```
import itertools as it

lst1 = [1, 2, 3]
lst2 = ['a', 'b', 'c']

res = it.chain(lst1, lst2)

print(list(res))
```

в стандартный поток вывода будет выведено:

- А. [1, 2, 3, 'a', 'b', 'c']
- Б. [1, 2, 3], ['a', 'b', 'c']
- В. [[1, 2, 3], ['a', 'b', 'c']]
- Г. нет правильного ответа

Ответ:

- А. [1, 2, 3, 'a', 'b', 'c']

8

В результате выполнения следующего кода

```
import itertools as it

lst1 = [1, 2, 3]
lst2 = ['a', 'b', 'c']

res = it.chain.from_iterable([lst1, lst2])

print(list(res))
```

в стандартный поток вывода будет выведено:

- А. [1, 2, 3, 'a', 'b', 'c']
- Б. [1, 2, 3], ['a', 'b', 'c']
- В. [[1, 2, 3], ['a', 'b', 'c']]
- Г. нет правильного ответа

Ответ:

- А. [1, 2, 3, 'a', 'b', 'c']

9	<p><i>В результате выполнения следующего кода</i></p> <pre>import itertools as it lst = [1, 2, 3, 4, 5, 6] res = it.islice(lst, 2) print(list(res))</pre> <p><i>в стандартный поток вывода будет выведено:</i></p> <p>А. [1, 2] Б. [5, 6] В. <itertools.islice object at 0x7fe68b07a3b0> (адрес в памяти будет другим) Г. нет правильного ответа</p> <p>Ответ: А. [1, 2]</p>
10	<p><i>В результате выполнения следующего кода</i></p> <pre>import itertools as it lst = [1, 5, 0, 1, 5] res = it.dropwhile(lambda x: x % 2 == 1, lst) print(list(res))</pre> <p><i>в стандартный поток вывода будет выведено:</i></p> <p>А. [1, 5, 0] Б. [1, 5] В. [0, 1, 5] Г. нет правильного ответа</p> <p>Ответ: В. [0, 1, 5]</p>
<p>ОПК-5. Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем</p>	
1	<p><i>Приведенный фрагмент файла</i></p> <pre>{ "journals": [{ "journalTitle": "Nature", "journalQuartile": "Q1" }, { "journalTitle": "Science", "journalQuartile": "Q1" }, { "journalTitle": "SoftwareX", "journalQuartile": "Q4" }]}</pre> <p><i>является примером разметки (формата)</i></p> <p>А. csv-файла Б. json-файла В. xml-файла Г. нет правильного ответа</p>

	<p>Ответ: Б. json-файла</p>
2	<p><i>Приведенный фрагмент файла</i></p> <pre>"Index", "Girth (in)", "Height (ft)", "Volume(ft^3)" 1, 8.3, 70, 10.3 2, 8.6, 65, 10.3 3, 8.8, 63, 10.2 4, 10.5, 72, 16.4 5, 10.7, 81, 18.8</pre> <p><i>является примером разметки (формата)</i></p> <p>А. csv-файла Б. json-файла В. xml-файла Г. нет правильного ответа</p> <p>Ответ: А. csv-файла</p>
3	<p><i>Следующий код</i></p> <pre>import itertools as it lst1 = [1, 2, 3] lst2 = ['a', 'b'] for val in it.product(lst1, lst2): print(val)</pre> <p><i>напечатает</i></p> <p>А. сочетания Б. размещения без повторений В. размещения с повторениями Г. декартово произведение</p> <p>Ответ: Г. декартово произведение</p>
4	<p><i>В результате выполнения следующего кода</i></p> <pre>import functools as ft @ft.lru_cache def f(x): return x * x print(f.cache_info())</pre> <p><i>будет</i></p> <p>А. сгенерировано исключение <code>AttributeError</code>, т.к. у <code>f</code> отсутствует атрибут <code>cache_info</code> Б. в стандартный поток вывода выведено <code>CacheInfo(hits=0, misses=0, maxsize=128, currsize=0)</code> В. в стандартный поток вывода выведено <code>CacheInfo(hits=0, misses=0, maxsize=None, currsize=0)</code> Г. нет правильного ответа</p>

	<p>Ответ: Б. в стандартный поток вывода выведено CacheInfo(hits=0, misses=0, maxsize=128, currsize=0)</p>
5	<p><i>В результате выполнения следующего кода</i></p> <pre>import functools as ft @ft.lru_cache def f(x): res = [x] return res val = f(1) val.append('A') print(f(1))</pre> <p><i>в стандартный поток вывода будет выведено</i></p> <p>А. [1] Б. [1, 'A'] В. ['A'] Г. нет правильного ответа</p> <p>Ответ: Б. [1, 'A']</p>
6	<p><i>В результате выполнения следующего кода</i></p> <pre>import functools as ft lst = [1, 2, 3, 4] res = ft.reduce(lambda x, y: x * y, lst) print(res)</pre> <p><i>в стандартный поток вывода будет выведено</i></p> <p>А. [1, 2, 6, 24] Б. [24] В. 24 Г. нет правильного ответа</p> <p>Ответ: В. 24</p>
7	<p><i>В результате выполнения следующего кода</i></p> <pre>import functools as ft lst = [1, 2, 3, 4] res = ft.reduce(lambda x, y: x + y, lst, 1014) print(res)</pre> <p><i>в стандартный поток вывода будет выведено</i></p> <p>А. [1015, 1017, 1020, 1024] Б. [1024] В. 1024 Г. нет правильного ответа</p>

	<p>Ответ: В. 1024</p>
8	<p><i>В результате выполнения следующего кода</i></p> <pre>import functools as ft @ft.singledispatch def scoo(x: str): print('My name is Scoo!') @scoo.register def by(x: int): print('My name is By') @scoo.register(type(None)) def doo(x): print('My name is Doo') scoo(1)</pre> <p><i>в стандартный поток вывода будет выведено</i></p> <p>А. My name is Scoo Б. My name is By В. My name is Doo Г. нет правильного ответа</p> <p>Ответ: Б. My name is By</p>
9	<p>@singledispatch из модуля functools выполняет определение вызываемой функции:</p> <p>А. по типу первого аргумента функции Б. по типам всех аргументов функции В. по типу первого аргумента функции и типу возвращаемого значения Г. по типам всех аргументов функции и типу возвращаемого значения</p> <p>Ответ: А. по типу первого аргумента функции</p>

10	<p><i>Дан следующий код.</i></p> <pre>import functools as ft pow = lambda x, y: x ** y pow2 = ft.partial(pow, 2) print(pow2(10))</pre> <p><i>При его выполнении</i></p> <p>А. будет порождено исключение <code>TypeError</code> из-за недостаточно числа переданных в функцию значений</p> <p>Б. будет напечатано значение 1024</p> <p>В. будет напечатано значение 1000</p> <p>Г. нет правильного ответа</p> <p>Ответ: В. будет напечатано значение 1000</p>
<p><i>ОПК-6. Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования</i></p>	
1	<p><i>Вы пытаетесь максимально полно утилизировать (использовать) имеющиеся у Вас в распоряжении вычислительные ядра CPU. Для этой цели Вы НЕ БУДЕТЕ использовать:</i></p> <p>А. <code>asyncio</code></p> <p>Б. <code>_thread</code></p> <p>В. <code>threading</code></p> <p>Г. нет правильного ответа</p> <p>Ответ: А. <code>asyncio</code></p>
2	<p><i>В Python 3.9 в <code>functools</code> был добавлен декоратор <code>@functools.cache(user_function)</code>. Прине его использования может выглядеть так:</i></p> <pre>import functools @functools.cache def cached_fun(x): ...</pre> <p><i>Какой декоратор Вы будете использовать в Python 3.7, чтобы обеспечить эквивалентное поведение:</i></p> <p>А. <code>@functools.lru_cache</code></p> <p>Б. <code>@functools.lru_cache(maxsize=256)</code></p> <p>В. <code>@functools.lru_cache(maxsize=None)</code></p> <p>Г. нет правильного ответа</p>

	<p>Ответ: В. @functools.lru_cache(maxsize=None)</p>
<p>3</p>	<p><i>Дан следующий код</i></p> <pre data-bbox="308 338 1046 875"> import time import _thread as _th def say_hello(name): time.sleep(1000) print(f'Hello, {name}') def run_all(): _th.start_new_thread(say_hello, ('Ann',)) _th.start_new_thread(say_hello, ('Hanna',)) print('All is done') run_all() </pre> <p><i>В результате его выполнения в стандартный поток вывода будет выведено:</i></p> <p>А. All is done</p> <p>Б. Hello, Ann Hello, Hanna All is done</p> <p>В. Hello, Hanna Hello, Ann All is done</p> <p>Г. нет правильного ответа</p> <p>Ответ: А. All is done</p>
<p>4</p>	<p><i>Выберите верное утверждение</i></p> <p><i>При использовании _thread:</i></p> <p>А. родительский поток ожидает завершения работы всех дочерних потоков (порожденных с помощью start_new_thread), после чего завершается сам</p> <p>Б. родительский поток завершается без ожидания завершения работы дочерних потоков (порожденных с помощью start_new_thread), но они продолжают свое нормальное исполнение в фоне (пока не выполнятся)</p> <p>В. при завершении родительского потока порожденные (с помощью start_new_thread) этим процессом дочерние потоки в широко распространенных системах принудительно завершаются</p> <p>Г. нет правильного ответа</p>

	<p>Ответ: В. при завершении родительского поток порожденные (с помощью <code>start_new_thread</code>) этим процессом дочерние потоки в широко распространенных системах принудительно завершаются</p> <p>Примечание: реализация поведения дочерних потоков в этом случае оставлена на усмотрение разработчиков, однако, в распространенных <code>unix</code>-подобных системах и <code>Windows</code> в этом случае дочерние потоки принудительно завершаются.</p>
5	<p>Дан следующий код.</p> <pre data-bbox="308 521 1086 1021"> import time import _thread as _th def say_hello(name): print(f'Hello, {name}') def run_all(): _th.start_new_thread(say_hello, ('Ann',)) _th.start_new_thread(say_hello, ('Hanna',)) run_all() </pre> <p>В нем для вывода в стандартный поток вывода мы использовали функцию <code>print</code>. Выберите верное утверждение:</p> <p>А. стандартный поток вывода является потокобезопасным, поэтому в него сначала будет выведено <code>'Hello, Ann'</code>, а затем – <code>'Hello, Hanna'</code></p> <p>Б. стандартный поток вывода является потокобезопасным, поэтому в него будут выведены обе фразы, но предсказать порядок, в котором они будут напечатаны, невозможно (это зависит от порядка выполнения потоков)</p> <p>В. стандартный поток вывода НЕ является потокобезопасным, поэтому предсказать результаты печати невозможно (строки могут быть «перемешаны» в случайном порядке)</p> <p>Г. нет правильного ответа</p> <p>Ответ: В. стандартный поток вывода НЕ является потокобезопасным, поэтому предсказать результаты печати невозможно (строки могут быть «перемешаны» в случайном порядке)</p>
6	<p>В рассмотренных выше примерах мы употребляем термин потоки как для стандартного потока вывода, так и для главного (родительского) и порожденного (дочернего) потоков исполнения. Выберите верное утверждение:</p> <p>Термины «поток» («дочерний поток» vs «стандартный поток вывода»)</p> <p>А. не имеют между собой ничего общего</p> <p>Б. в обоих случаях обозначают один и тот же объект (экземпляр класса)</p> <p>В. в обоих случаях обозначают один и тот же класс, но разные объекты (экземпляры класса)</p> <p>Г. нет правильного ответа</p>

	<p>Ответ: А. не имеют между собой ничего общего</p> <p>Примечание: в данном случае мы имеем не совсем удачный (хотя и устоявшийся) перевод английских терминов «thread» и «stream», оба из которых в русскоязычной IT-литературе переводятся как «поток».</p>
7	<p><i>Для синхронизации параллельных обращений из разных потоков к разделяемому ресурсу можно использовать (выберите верное утверждение):</i></p> <p>А. инструкцию <code>time.sleep(10)</code> Б. инструкцию <code>time.sleep</code>, но ее параметр (аргумент) следует определять из решаемой задачи («нужно только правильно слипы подобрать») В. блокировки Г. нет правильного ответа</p> <p>Ответ: В. блокировки</p>
8	<p>Выберите верное утверждение:</p> <p>А. библиотека <code>_thread</code> использует библиотеку <code>threading</code> в своей реализации Б. библиотека <code>threading</code> использует библиотеку <code>_thread</code> в своей реализации В. обе библиотеки (<code>threading</code> vs <code>_thread</code>) друг от друга независимы Г. нет правильного ответа</p> <p>Ответ: Б. библиотека <code>threading</code> использует библиотеку <code>_thread</code> в своей реализации</p>
9	<p><i>Выберите верное утверждение:</i></p> <p>А. класс <code>Queue</code> библиотеки <code>queue</code> предоставляет потокобезопасный API для добавления и получения элементов, причем обеспечивается принцип «первым вошел – первым вышел» Б. класс <code>Queue</code> библиотеки <code>queue</code> предоставляет потокобезопасный API для добавления и получения элементов, причем обеспечивается принцип «первым вошел – последним вышел» В. класс <code>Queue</code> библиотеки <code>queue</code> НЕ предоставляет потокобезопасный API для добавления и получения элементов, поэтому программисту необходимо самостоятельно управлять (выполнять синхронизацию) параллельным доступом, чтобы избежать повреждения состояния объекта (экземпляра класса) <code>Queue</code>. Г. нет правильного ответа</p> <p>Ответ: Б. класс <code>Queue</code> библиотеки <code>queue</code> предоставляет потокобезопасный API для добавления и получения элементов, причем обеспечивается принцип «первым вошел – последним вышел»</p>

10

Выберите **НЕ** верное утверждение:

Библиотека `asyncio`

- А. предназначена для написания асинхронного кода, который выглядит как синхронный
- Б. предназначена для написания синхронного кода, который выглядит как асинхронный
- В. использует модель «однопоточный цикл событий»
- Г. нет правильного ответа

Ответ:

- Б. предназначена для написания синхронного кода, который выглядит как асинхронный