ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Вычислительной и прикладной математики»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ
«Учебная практика»

Направление подготовки
09.03.03 «Прикладная информатика»

Направленность (профиль) подготовки
Прикладная информатика в экономике

Квалификация выпускника – бакалавр

Форма обучения – очная, заочная

Рязань

1. ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур), предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной профессиональной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям ОПОП.

Основная задача – обеспечить оценку уровня сформированности общекультурных, общепрофессиональных и профессиональных компетенций.

Контроль знаний обучающихся проводится в форме промежуточной аттестации.

Промежуточный контроль по дисциплине осуществляется проведением зачета с оценкой.

2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;

2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;

3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Уровень освоения компетенций, формируемых дисциплиной:

а) описание критериев и шкалы оценивания тестирования:

Шкала оценивания	**Критерий**
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 75 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 60 до 74%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 59%

б) описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	**Критерий**
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя.
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов.
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя.
0 баллов	выставляется студенту, который не смог ответить на вопрос

в) описание критериев и шкалы оценивания практического задания:

Шкала оценивания	**Критерий**
3 балла	Задача решена верно

(эталонный уровень)	
2 балла (продвинутый уровень)	Задача решена верно, но имеются технические неточности в расчетах
1 балл (пороговый уровень)	Задача решена верно, с дополнительными наводящими вопросами преподавателя
0 баллов	Задача не решена

На зачет с оценкой выносится: тестовое задание, 1 практическое задание и 1 теоретический вопрос. Студент может набрать максимум 9 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Шкала оценивания	**Критерий**	
отлично (эталонный уровень)	8 – 9 баллов	Обязательным условием является выполнение всех предусмотренных в течение семестра заданий
хорошо (продвинутый уровень)	6 – 7 баллов	
удовлетворительно (пороговый уровень)	4 – 5 баллов	
неудовлетворительно	0 – 3 баллов	Студент не выполнил всех предусмотренных в течение семестра текущих заданий

3. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

№ п/п	**Контролируемые разделы (темы) дисциплины** **(результаты по разделам)**	**Код контролируемой компетенции (или её части)**	**Наименование оценочного мероприятия**
1	2	3	4
1	Тема 1. Среда программирования и структура программы на языке Питон	УК-1.1 ОПК-2.1 ОПК-2.2 ОПК-7.1 ОПК-7.2 ОПК-7.3	Зачет с оценкой
2	Тема 2. Программирование разветвляющихся алгоритмов	УК-1.1 ОПК-2.1 ОПК-2.2 ОПК-7.1 ОПК-7.2 ОПК-7.3	Зачет с оценкой
3	Тема 3. Программирование циклических алгоритмов	УК-1.1 ОПК-2.1 ОПК-2.2 ОПК-7.1 ОПК-7.2 ОПК-7.3	Зачет с оценкой
4	Тема 4. Программирование практических задач	УК-1.1 ОПК-2.1 ОПК-2.2 ОПК-7.1 ОПК-7.2 ОПК-7.3	Зачет с оценкой

4. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1. Промежуточная аттестация (зачет с оценкой)

УК-1: Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК-1.1. Осуществляет поиск необходимой информации, подвергает ее критическому анализу и обобщению

ОПК-2: Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решения задач профессиональной деятельности;
ОПК-2.1. Понимает состояние и тенденции развития современных информационных технологий и программных средств, в том числе отечественного производства
ОПК-2.2. Использует при решении задач профессиональной деятельности современные информационные технологии и программные средства, в том числе отечественного производства
ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического использования
ОПК-7.1. Понимает требования к алгоритмам, суть процесса алгоритмизации задач
ОПК-7.2. Выполняет разработку алгоритмического и программного обеспечения для решения прикладных задач.
ОПК-7.3. Применяет основы информатики и программирования к проекту, конструирует и тестирует программный продукт

Вопросы к зачету по учебной практике

1. Что такое Python?
2. В каком году написана первая статья про Python?
3. Какие типы данных есть в Python? На какие классы делятся?
4. Что такое лямбда-функция? Какое у неё назначение?
5. Что такое PEP 8?
6. Как получить документацию по атрибутам объекта?
7. Что такое docstring?
8. В чём разница между типами list и tuple?
9. Может ли быть индекс списка отрицательным?
10. Что значит конструкция pass?
11. Чем отличаются многопоточное и многопроцессорное приложение?
12. Как просмотреть методы объекта?
13. Что такое *args и **kwargs в определении функции?
14. Python полностью поддерживает ООП?
15. Что такое globals() и locals()?
16. Что хранится в атрибуте __dict__?
17. Как проверить файл .py на синтаксические ошибки, не запуская его?
18. Зачем в Python используется ключевое слово self?
19. Что такое декоратор? Как написать собственный?
20. Что может быть ключом в словаре?
21. В чём разница между пакетами и модулями?
22. Как перевести строку, содержащую двоичный код (1 и 0), в число?
23. Для чего используется функция __init__?
24. Что такое слайс(slice)?
25. Как проверить, что один кортеж содержит все элементы другого кортежа?
26. Почему пустой список нельзя использовать как аргумент по умолчанию?
27. Что такое @classmethod, @staticmethod, @property?
28. Что такое синхронный код?
29. Что такое асинхронный код? Приведите пример.
30. Каким будет результат следующего выражения?

```
-30 % 10
```

31. Для чего нужен метод id()?
32. Что такое итератор?
33. Что такое генератор? Чем отличается от итератора?
34. Для чего используется ключевое слово yield?
35. Чем отличаются __iter__ и __next__?

36. Что такое контекстный менеджер?
37. Как сделать Python-скрипт исполняемым в различных операционных системах?
38. Как сделать копию объекта? Как сделать глубокую копию объекта?
39. Опишите принцип работы сборщика мусора в Python
40. Как использовать глобальные переменные? Это хорошая идея?
41. Для чего в классе используется атрибут __slots__?
42. Какие пространства имен существуют в Python?
43. Как реализуется управление памятью в Python?
44. Что такое метаклассы и в каких случаях их следует использовать?
45. Зачем нужен pdb?
46. Каким будет результат следующего выражения?

```
[0, 1][10:]
```

47. Как создать класс без слова class?
48. Как перезагрузить импортированный модуль?
49. Напишите декоратор, который будет перехватывать ошибки и повторять функцию максимум N раз.
50. Каким будет результат следующего выражения?

```
len(' '.join(list(map(str, [[0], [1]]))))
```

51. Python — легкий язык. Согласны?
52. Какие проблемы есть в Python?
53. Когда будет выполнена ветка else в конструкции try...except...else?
54. Поддерживает ли Python множественное наследование?
55. Как dict и set реализованы внутри? Какова сложность получения элемента? Сколько памяти потребляет каждая структура?
56. Что такое MRO? Как это работает?
57. Как аргументы передаются в функции: по значению или по ссылке?
58. С помощью каких инструментов можно выполнить статический анализ кода?
59. Что будет напечатано в результате выполнения следующего кода?

```
import sys

arr_1 = []
arr_2 = arr_1
print(sys.getrefcount(arr_1))
```

60. Что такое GIL? Почему GIL всё ещё существует?
61. Опишите процесс компиляции в Python.
62. Как тиражировать Python код?
63. Что такое дескрипторы? Есть ли разница между дескриптором и декоратором?
64. Почему всякий раз, когда Python завершает работу, не освобождается вся память?
65. Что будет напечатано в результате выполнения следующего кода?

```
class Variable:
    def __init__(self, name, value):
        self._name = name
        self._value = value

    @property
    def value(self):
        print(self._name, 'GET', self._value)
        return self._value

    @value.setter
    def value(self, value):
        print(self._name, 'SET', self._value)
        self._value = value

var_1 = Variable('var_1', 'val_1')
var_2 = Variable('var_2', 'val_2')
var_1.value, var_2.value = var_2.value, var_1.value
```

66. Что такое интернирование строк? Почему это есть в Python?
67. Как упаковать бинарные зависимости?
68. Почему в Python нет оптимизации хвостовой рекурсии? Как это реализовать?
69. Что такое wheels и eggs? В чём разница?
70. Как получить доступ к модулю, написанному на Python из C и наоборот?
71. Как ускорить существующий код Python?
72. Что такое __pycache__? Что такое файлы .pyc?
73. Что такое виртуальное окружение?
74. Python — это императивный или декларативный язык?
75. Что такое менеджер пакетов? Какие менеджеры пакетов вы знаете?
76. В чём преимущества массивов NumPy по сравнению с (вложенными) списками Python?
77. Вам нужно реализовать функцию, которая должна использовать статическую переменную. Вы не можете писать код вне функции и у вас нет информации о внешних переменных (вне вашей функции). Как это сделать?
78. Что будет напечатано в результате выполнения следующего кода?

```
def f_g():
    yield 43
    return 66

print(f_g())
```

79. Как имплементировать словарь с нуля?
80. Напишите однострочник, который будет подсчитывать количество заглавных букв в файле.
81. Что такое файлы .pth?
82. Какие функции из Collections и Itertools вы используете?
83. Что делает флаг PYTHONOPTIMIZE?
84. Что будет напечатано в результате выполнения следующего кода?

```
arr = [[]] * 5
arr_1, arr_2 = arr, arr

for k, arr in enumerate((arr_1, arr_2)):
    arr[0].append(k)

arr = (arr_1, 5, arr_2)
print(arr)
```

85. Какие переменные среды, влияющие на поведение интерпретатора Python, вы знаете?
86. Что такое Cython? Что такое IronPython? Что такое PyPy? Почему они до сих пор существуют и зачем?
87. Как перевернуть генератор?
88. Приведите пример использования filter и reduce над итерируемым объектом.
89. Что будет напечатано в результате выполнения кода?

```
print(_)
```

90. Чем фреймворк отличается от библиотеки?
91. Расположите функции в порядке эффективности, объясните выбор.