


ФГБОУ ВО НОВОСИБИРСКИЙ ГАУ
Кафедра информационных технологий и моделирования

УТВЕРЖДЕН

Рег. № ПЧ.03-42
«05» 10 2022г.

на заседании кафедры
Протокол от «23» 09 2022г. № 2
Заведующий кафедрой информационных
технологий и моделирования


(подпись) О.В. Агафонова

ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ

Б1.В.12 Программирование на Python

Шифр и наименование дисциплины

09.03.03 Прикладная информатика

Код и наименование направления подготовки

Прикладная информатика

Направленность (профиль)

Новосибирск 2022

Паспорт фонда оценочных средств

№ п/п	Наименование разделов и тем	Код контролируемо й компетенции (или ее части)	Наименование оценочного средства
1.	<i>Введение. Основы объектно-ориентированного подхода</i>	ПК-2	Вопросы для собеседования №1
2.	<i>Основы программирования на объектно-ориентированном языке Python</i>		
2.1	Основные типы данных Python	ПК-2	Вопросы для коллоквиума №1
2.2	Базовые конструкции языка	ПК-2	Вопросы для коллоквиума № 2
2.3	Функции	ПК-2	Разноуровневые задания 1
2.4	Файлы	ПК-2	Разноуровневые задания 2
2.5	Классы	ПК-2	Разноуровневые задания 3
3.	<i>Профессиональный уровень программирования на объектно-ориентированном языке Python</i>		
3.1	Визуализация данных	ПК-2	Разноуровневые задания 4
3.2	Работа с данными в форматах CSV и JSON	ПК-2	Вопросы для коллоквиума № 3
3.3	Использование API для получения данных сайтов	ПК-2	Вопросы для коллоквиума № 4
	Контрольная работа, экзамен	ПК-2	Задание для контрольной работы, вопросы к экзамену

Вопросы для собеседования №1

Тема: «Введение. Основы объектно-ориентированного подхода»

1. Принципы, протоколы и стандарты объектного подхода.
2. Объектная декомпозиция, виды абстракций (классы).
3. Объектная декомпозиция, представители классов - объекты.
4. Объекты, понятие, назначение.
5. Объекты, свойства и методы, связи между собой.
6. Типы отношений между классами.
7. Объектно-ориентированный язык программирования Python, характеристика, возможности, конкурентоспособность.
8. Визуальная среда программирования.
9. Визуальная среда программирования, назначение, характеристика. Визуальная среда программирования, виды. Оценка.
10. Интерфейс среды программирования Pycharm, порядок работы, создание проекта, разработка и отладка.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума №1
Тема: «Основные типы данных Python»

1. Основные типы данных Python. Обзор.
2. Простые типы данных.
3. Числовой тип данных. Их характеристики и операции.
4. Строковый тип данных. Их характеристики и операции.
5. Переменные. Их характеристики и операции.
6. Списки. Их характеристики и операции.
7. Кортежи. Их характеристики и операции.
8. Словари. Их характеристики и операции.
9. Множества. Их характеристики и операции.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 2
Тема: «Базовые конструкции языка»

1. Базовые конструкции языка основных элементов.
2. Основные операторы, характеристики.
3. Основные операторы, виды.
4. Оператор присваивания. Порядок применения.
5. Оператор обращения к процедуре. Порядок применения.
6. Структурные операторы. Порядок применения.
7. Составной оператор. Порядок применения.
8. Условные операторы. Порядок применения.
9. Операторы цикла. Порядок применения.
10. Оператор присоединения. Порядок применения.
11. Условный оператор if. Порядок применения.
12. Операторы циклов for. Порядок применения.
13. Операторы циклов while. Порядок применения.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Разноуровневые задания 1

Тема: «Функции в Python. Создание функций»

1. Задания репродуктивного уровня

Работа по примеру.

Пример функции без параметров.

```
deff_print_ok(): print('всеуспешно! ')
```

Пример функции с параметром.

```
deff_print(m):  
    print(m)  
deff_sum(x,y):  
    return x+y
```

Вызов функции.

```
f_print_ok()  
f_print(2345)  
f_print('Питон')  
z = f_sum(10,20)  
print('z=', z) # z= 30  
z = f_sum('str', str(20))  
print('z=', z) # z= str20  
print(f_sum('str', str(1120))) # str1120
```

Сохранение ссылки на функции в переменной.

```
deff_sum(x,y):  
    return x+y  
z = f_sum # в переменную z записываем ссылку на функцию  
print(type(z)) # <class 'function'>  
rez = z(15,40)  
print('rez=', rez) # rez= 55
```

Функции обратного вызова - функции, передаваемые по ссылке.

```
deff_sum(x,y):  
    return x+y  
def f_sum2(f, a, b):  
    return f(a, b) # через f будет доступна ссылка на ф-ю f_sum()  
v = f_sum2(f_sum, 40, 10)  
print('v=', v) # v= 50
```

```

z = f_sum      # в переменную z записываем ссылку на функцию
f_sum()
v = f_sum2(z, 40, 70)
print('v=', v) # v= 110

```

2. Задания реконструктивного уровня

Работа по примеру.

Определение функции в зависимости от условия.

```

m = input('Введите 1 для вызова 1-й функции:')
if m == '1':
    def f_print():
        return 'Вы ввели число 1'
else:
    def f_print():
        return 'Альтернативная функция'
print(f_print())

```

Чтобы сделать параметр необязательным, нужно присвоить ему начальное значение при объявлении ф-и. Необязательные параметры должны следовать после обязательных.

```

def f_sum(x, y = 2):    # y - необязательный параметр
    return x+y
v1 = f_sum(5)
print('v1=', v1) # v1= 7
v2 = f_sum(10, 20)
print('v2=', v2) # v2= 30

```

Сопоставление по ключам - при вызове функции присваиваются значения.

```

def f_sum(x, y):
    return x+y
print('v3=', f_sum(x = 16, y = 74))    # v3 = 90

```

Сопоставление по ключам удобно использовать, если функция имеет несколько необязательных параметров.

```

def f_sum(a = 2, b = 3, c = 4):    # все параметры необязательные
    return a+b+c
print('r1=', f_sum(2, 3, 20))    # позиционное присваивание r1= 25
print('r2=', f_sum(c = 20))    # сопоставление по ключам r2= 25

```

Передача параметров значениями из кортежей и списков.

```
def f_sum(a, b, c):  
    return a+b+c  
t1, arr = (1, 2, 3), [1, 2, 3]  
print('t1:', f_sum(*t1))    # передача параметров кортежем t1: 6  
print('arr:', f_sum(*arr))  # передача параметров списком arr: 6  
t2 = (2, 3)  
print('t2:', f_sum(1, *t2))    # значения можно комбинировать t2: 6
```

Передача значений параметров из словаря:

```
def f_sum(a, b, c):  
    return a+b+c
```

3. Задания творческого уровня

Лямбда-функции.

```
f1 = lambda: 10+20 # функция без параметров  
f2 = lambda x, y: x+y # функция с двумя параметрами  
f3 = lambda x, y, z: x+y+z # функция с тремя параметрами  
print(f1()) # 30  
print(f2(10, 20)) # 30  
print(f3(10, 20, 30)) # 60
```

Необязательным параметрам в лямбда-функциях присваивается значение по умолчанию.

```
f = lambda x, y = 2: x+y  
print(f(5)) # 7  
print(f(5, 6)) # 11
```

Критерии оценки:

- оценка «отлично» выставляется студенту, если выполнены задания репродуктивного, реконструктивного и творческого уровня.
- оценка «хорошо» выставляется студенту, если выполнены задания репродуктивного, реконструктивного и некоторые задания творческого уровня.
- оценка «удовлетворительно» выставляется студенту, если выполнены задания репродуктивного, реконструктивного уровня.
- оценка «неудовлетворительно» выставляется студенту, если выполнены задания репродуктивного уровня.

Разноуровневые задания 2

Тема: «Файлы»

1. Задания репродуктивного уровня

Работа по примеру.

Создание нового текстового файла или открытие существующего файла на запись:

```
f = open('fruit.txt', 'w')
добавить в файл строки по одной:
f.write('яблоки\n')
f.write('груши\n')
f.write('персики\n')
f.write('сливы\n')
добавить в файл несколько строк в цикле:
z = 1
while z < 5:
    name = input('Введите название фрукта:')
    f.write(name+'\n')
    z += 1
закрыть файл:
f.close()
открыть файл на чтение:
f = open('fruit.txt', 'r')
распечатать весь файл:
lines = f.read()
print(lines)
f.close()
input()
```

2. Задания реконструктивного уровня

Прочитать текстовый файл в список, элементами которого являются строки:

```
f = open('fruit.txt', 'r')
lines = f.readlines()
print(lines)
f.close()
input()
дополнить текстовый файл новыми строками и снова распечатать:
f = open('fruit.txt', 'a')
for i in range(3):
```

```
name = input('Введите новое название фрукта:')  
f.write(name+'\n')  
f.close()  
f = open('fruit.txt', 'r')  
lines = f.read()  
print(lines)  
f.close()
```

3. Задания творческого уровня

Задание.

1. Создайте новый текстовый файл Capitals и введите в него названия 5 столиц мира. Выведите файл на печать.
2. Сформируйте из файла список и распечатайте.
3. Дополните файл еще 5 названиями мировых столиц. Распечатайте файл.

Критерии оценки:

- оценка «отлично» выставляется студенту, если выполнены задания репродуктивного, реконструктивного и творческого уровня.
- оценка «хорошо» выставляется студенту, если выполнены задания репродуктивного, реконструктивного и некоторые задания творческого уровня.
- оценка «удовлетворительно» выставляется студенту, если выполнены задания репродуктивного, реконструктивного уровня.
- оценка «неудовлетворительно» выставляется студенту, если выполнены задания репродуктивного уровня.

Разноуровневые задания 2

Тема: «Классы»

1. Задания репродуктивного уровня

1. Создание класса. Определение атрибутов и методов объектов класса.

Создание класса Dog(), моделирующего объект Собака

```
class Dog():
```

```
    """ Модель просто собаки """
```

```
    def __init__(self, name, age):
```

```
        """ Описание и инициализация атрибутов name и age """
```

```
        self.name = name
```

```
        self.age = age
```

```
        """ Описание методов """
```

```
    def voice(self):
```

```
        """ Собака лает по команде """
```

```
        print(f"{self.name} по команде лает")
```

```
    def paw(self):
```

```
        """ Собака дает лапу по команде """
```

```
        print(f"{self.name} по команде дает лапу")
```

2. Задания реконструктивного уровня

2. Создание объекта класса. Определение значений атрибутов объекта, применение методов

"""Создание экземпляра класса Dog(), представляющего конкретную собаку. Кличка собаки – Джим, возраст собаки – 3 года """

```
my_dog = Dog('Джим', 3)
```

```
""" Обращение к атрибутам класса Dog() """
```

```
print(f"Кличка моей собаки {my_dog.name}.")
```

```
print(f"Возраст моей собаки {my_dog.age} yearsold.")
```

```
""" Вызов методов класса Dog() """
```

```
my_dog.sit()
```

```
my_dog.roll_over()
```

```
my_dog.voice()
```

```
my_dog.paw()
```

Создание нескольких экземпляров класса Dog()

```

my_dog = Dog('Джим', 3)
your_dog = Dog('Веста', 2)
print(f"Кличка моей собаки {my_dog.name}.")
print(f"Возраст моей собаки {my_dog.age} years old.")
print(f"Кличка твоей собаки { your_dog.name}.")
print(f"Возраст твоей собаки { your_dog.age} years old.")
my_dog.voice() "моя собака лает "
my_dog.paw() " моя собака дает лапу "
your_dog.voice() "твоя собака лает"
your_dog.paw() "твоя собака дает лапу"

```

3. Задания творческого уровня

3. Вывод отформатированных значений атрибутов

```

class Dog():
    """ Простая модель собаки """
    def __init__(self, name, age, poroda):
        """ Инициализирует атрибуты name, age, poroda """
        self.name = name
        self.age = age
        self.poroda = poroda
    def get_descriptive_dog(self):
        """ вывод отформатированного описания собаки """
        all_name = f"Сведения о моей собаке: {self.name} {self.age} года {self.poroda}"
        return all_name.title()
my_dog = Dog('Джим', 3, 'терьер')
print(my_dog.get_descriptive_dog())

```

Критерии оценки:

- оценка «отлично» выставляется студенту, если выполнены задания репродуктивного, реконструктивного и творческого уровня.
- оценка «хорошо» выставляется студенту, если выполнены задания репродуктивного, реконструктивного и некоторые задания творческого уровня.
- оценка «удовлетворительно» выставляется студенту, если выполнены задания репродуктивного, реконструктивного уровня.
- оценка «неудовлетворительно» выставляется студенту, если выполнены задания репродуктивного уровня.

Тема: «Визуализация данных»

1. Задания репродуктивного уровня

Инструменты визуализации данных. Библиотека matplotlib.

Подключение matplotlib:

```
importmatplotlib.pyplot as plt
```

Построение простого графика.

```
importmatplotlib.pyplot as plt
```

```
squares = [2, 75, 89, 4, 35, 15]
```

```
fig,ax = plt.subplots()
```

```
ax.plot(squares)
```

```
plt.show()
```

2. Задания реконструктивного уровня

Изменение типа надписей и толщины графика.

```
importmatplotlib.pyplot as plt
```

```
squares = [2, 75, 89, 4, 35, 15]
```

```
fig,ax = plt.subplots()
```

```
ax.plot(squares,linewidth = 7)
```

```
plt.show()
```

Назначение заголовка диаграммы, меток осей, размера шрифта делений на осях.

```
importmatplotlib.pyplot as plt
```

```
squares = [2, 75, 89, 4, 35, 15]
```

```
fig,ax = plt.subplots()
```

```
ax.plot(squares, linewidth = 7)
```

```
ax.set_title("Square Numbers", fontsize = 24)
```

```
ax.set_xlabel("Value",fontsize = 14)
```

```
ax.set_ylabel("Square of Value", fontsize = 14)
```

```
ax.tick_params(axis = 'both', labelsizes = 14)
```

```
plt.show()
```

Встроенные стили оформления диаграмм.

Запустите код:

```
importmatplotlib.pyplot as plt
```

```
print("Встроенные стили оформления диаграмм:")
```

```
print(plt.style.available)
```

Установить стиль диаграммы в программе:

```
plt.style.use('grayscale')
plt.style.use('seaborn-darkgrid')
```

Нанесение и оформление отдельных точек. Функция `scatter()`

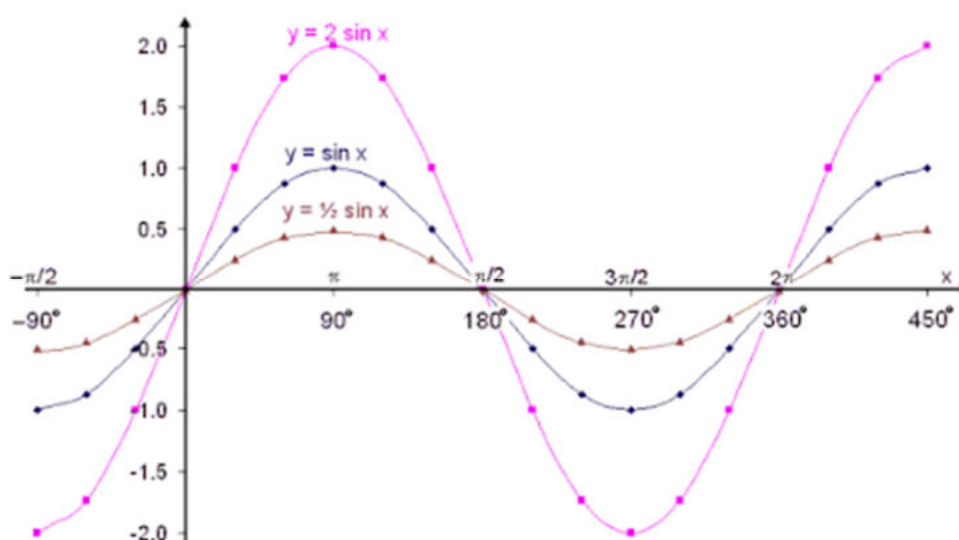
```
plt.style.use('Solarize_Light2')
```

```
ax.scatter(2, 15) # 2, 15 – координаты точки
```

```
ax.scatter(2,15, s = 400) # увеличим размер точки
```

3. Задания творческого уровня

Нарисуйте как можно более похожую картинку в `matplotlib`. Добавьте к ней описание осей, сетку, легенду, название. Сохраните картинку из Jupyter или Collab и отправьте преподавателю.



Критерии оценки:

– оценка «отлично» выставляется студенту, если выполнены задания репродуктивного, реконструктивного и творческого уровня.

– оценка «хорошо» выставляется студенту, если выполнены задания репродуктивного, реконструктивного и некоторые задания творческого уровня.

– оценка «удовлетворительно» выставляется студенту, если выполнены задания репродуктивного, реконструктивного уровня.

– оценка «неудовлетворительно» выставляется студенту, если выполнены задания репродуктивного уровня.

Вопросы для коллоквиума №3

Тема: «Работа с данными в форматах CSV и JSON»

1. Работа с данными в форматах CSV. Назначение.
2. Работа с данными в форматах JSON. Назначение.
3. Формирование массивов для хранения данных.
4. Формат данных JSON, основные операции.
5. Формат данных CSV, основные операции.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Вопросы для коллоквиума № 4

Тема «Использование API для получения данных сайтов»

1. Использование для получения данных сайтов.API, понятие.
2. Использование для получения данных сайтов.API, назначение.
3. Вызовы API.
4. API веб-приложения.
5. Библиотека Requests, назначение.
6. Визуализация данных, полученных в результате вызова API

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно дан ответ на поставленный вопрос;
- оценка «хорошо» если правильно дан ответ на поставленный вопрос после дополнительного вопроса преподавателя;
- оценка «удовлетворительно», если правильно дан ответ на поставленный вопрос после нескольких дополнительных вопросов преподавателя;
- оценка «неудовлетворительно», если студент не ответил на поставленный вопрос.

Задание для контрольной работы

Тема: «Создание и использование классов»

1. Создайте класс для описания объекта Спортсмен.
2. Класс должен содержать 5 атрибутов (ФИО, год рождения, вид спорта, число участия в соревнованиях, сколько раз занял призовое место).
3. Класс должен содержать 3-4 метода (например, участвует в соревнованиях, занял призовое место и т.д.)
4. Создайте 3 экземпляра класса Спортсмен.
5. Выведите на печать значения атрибутов и вызов методов для каждого экземпляра.
6. Напишите методы для изменения значений атрибутов число участия в соревнованиях и сколько раз занял призовое место способом непосредственного изменения значений.
7. Напишите код для изменения значений атрибутов число участия в соревнованиях и сколько раз занял призовое место способом новых методов, включенных в класс
8. Напишите код для изменения значений атрибутов число участия в соревнованиях и сколько раз занял призовое место способом новых методов с приращением, включенных в класс.

Критерии оценки

Для оценки работы вводится 20 балльная оценочная шкала. На заключительном этапе оценочная шкала переводится в «зачтено» или «не зачтено».

- 1) выполнение работы в соответствии с установленными требованиями к содержанию и оформлению: 16-18б;
- 2) выполнение работы с несущественными ошибками: 10-15б;
- 3) выполнение работы с грубыми ошибками: 1-9б;
- 4) сдача работы в установленные рабочими программами сроки: 2б.

Оценочная шкала для итоговой проверки работы заключается в следующем:

1. Для отметки «Зачтено» необходимо набрать свыше 15 баллов.
2. Для отметки «Не зачтено» - количество баллов от 0 до 15.

Вопросы к экзамену

1. Принципы, протоколы и стандарты объектного подхода
2. Объектная декомпозиция.
3. Виды абстракций (классы).
4. Представители классов - объекты.
5. Объекты, понятие, назначение, их свойства и методы.
6. Типы отношений между классами.
7. Объектно-ориентированный язык программирования Python, характеристика.
8. Объектно-ориентированный язык программирования Python, возможности, конкурентоспособность.
9. Визуальная среда программирования Pycharm
10. Визуальная среда программирования, назначение, характеристика.
11. Визуальная среда программирования, виды. Оценка.
12. Интерфейс среды программирования.
13. Визуальная среда программирования, порядок работы, создание проекта. Визуальная среда программирования, разработка и отладка программ.
14. Основные типы данных Python. Обзор.
15. Простые типы данных.
16. Числовой тип данных. Строковый тип данных. Их характеристики и операции.
17. Переменные. Списки. Их характеристики и операции.
18. Кортежи. Словари. Их характеристики и операции.
19. Базовые конструкции языка основных элементов.
20. Основные операторы, характеристики, виды.
21. Структурные операторы. Порядок применения.
22. Условные операторы. Порядок применения.
23. Операторы цикла. Порядок применения.
24. Функции. Создание.
25. Структура функции. Параметры функции.
26. Внутренние и внешние переменные.
27. Анонимные функции, назначение, способы использования.
28. Файлы, понятие, назначение.
29. Виды файлов. Операции с файлами.
30. Классы. Понятие, назначение
31. Структура класса.
32. Создание и использование классов.

33. Атрибуты класса, понятие, определение.
34. Способы изменения атрибутов.
35. Наследование классов, понятие, реализация.
36. Переопределение методов класса-родителя
37. Визуализация данных.
38. Инструменты визуализации данных.
39. Библиотека matplotlib.
40. Построение простого графика, порядок.
41. Оформление графика.
42. Работа с данными в форматах CSV. Назначение этих форматов
43. Работа с данными в форматах CS JSON. Назначение этих форматов. Формирование массивов для хранения данных.
44. Использование для получения данных сайтов. API, понятие, назначение. Вызовы API.
45. Библиотека Requests, назначение.
46. Визуализация данных, полученных в результате вызова API.

Критерии оценки:

Отметка **«ОТЛИЧНО»** - дан полный, развернутый ответ на поставленные вопросы, показана совокупность осознанных знаний по дисциплине; в ответе прослеживается четкая структура и логическая последовательность, отражающая сущность раскрываемых понятий. Ответ изложен литературным языком с использованием современной гистологической терминологии. Могут быть допущены недочеты в определении понятий, исправленные студентом самостоятельно в процессе ответа.

Отметка **«ХОРОШО»** - дан полный, развернутый ответ на поставленный вопросы. Ответ четко структурирован, логичен, изложен литературным языком с использованием современной гистологической терминологии. Могут быть допущены 2-3 неточности или незначительные ошибки, исправленные студентом с помощью преподавателя

Отметка **«УДОВЛЕТВОРИТЕЛЬНО»** - даны недостаточно полный и недостаточно развернутый ответы. Логика и последовательность изложения имеют нарушения. Допущены ошибки в раскрытии понятий, употреблении терминов. В ответе отсутствуют выводы. Умение раскрыть значение обобщенных знаний не показано. Речевое оформление требует поправок, коррекции.

Отметка «**НЕУДОВЛЕТВОРИТЕЛЬНО**» - ответ представляет собой разрозненные знания с существенными ошибками по вопросам. Присутствуют фрагментарность, нелогичность изложения. Отсутствуют выводы, конкретизация и доказательность изложения. Речь неграмотная, гистологическая терминология не используется. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента. Или ответ на вопрос полностью отсутствует, или отказ от ответа.

ЗАДАНИЯ ДЛЯ ОЦЕНКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Задания для оценки уровня сформированности компетенции ПК-2

1. С какими типами данных работает Python (выберете неверный вариант)

- a. переменная
- b. словарь
- c. список
- d. строки
- e. совокупность
- f. числа
- g. множества

Ответ: e

2. Какие типы данных относятся к изменяемым типам данных? (множественный выбор)

- a. переменная
- b. словарь
- c. список
- d. строки
- e. числа
- f. множества

Ответ: b,c

3. Какие типы данных относятся к НЕ изменяемым типам данных? (множественный выбор)

- a. переменная
- b. словарь
- c. список
- d. строки
- e. числа
- f. множества

Ответ: a,d,e

4. Тип данных характеризует одновременно:

- a. множество допустимых значений, которые могут принимать данные, принадлежащие к этому типу;
- b. набор операций, которые можно осуществлять над данными, принадлежащими к этому типу;
- c. оба варианта.

Ответ: c

5. Данные какого типа заключаются в одинарные или двойные кавычки?

Ответ: ...

6. Этот тип данных используется для хранения числовых значений.

Ответ: ...

7. Перечислите числовые типы данных.

Ответ: ...

8. Перечислите последовательности, с которыми работает РУ.

Ответ: ...

Критерии оценки результатов тестирования:

– оценка «отлично» выставляется студенту, если он отвечает верно на 80-100% вопросов.

– оценка «хорошо», выставляется студенту, если он отвечает верно на 70-79% вопросов.

– оценка «удовлетворительно», выставляется студенту, если он отвечает верно на 60-69% вопросов.

– оценка «неудовлетворительно» выставляется студенту, если он не освоил материал темы, дает менее 60% правильных ответов.

МАТРИЦА СООТВЕТСТВИЯ КРИТЕРИЕВ ОЦЕНКИ УРОВНЮ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Критерии оценки	Уровень сформированности компетенций
Оценка по пятибалльной системе	
«Отлично»	«Высокий уровень»
«Хорошо»	«Повышенный уровень»
«Удовлетворительно»	«Пороговый уровень»
«Неудовлетворительно»	«Не достаточный»
Оценка по системе «зачет – незачет»	
«Зачтено»	«Достаточный»
«Не зачтено»	«Не достаточный»

Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

1. Положение «О балльно-рейтинговой системе аттестации студентов»: СМК ПНД 08-01-2022, введено приказом от 28.09.2011 №371-О (<http://nsau.edu.ru/file/403>: режим доступа свободный);

2. Положение «О проведении текущего контроля и промежуточной аттестации обучающихся в ФГБОУ ВО Новосибирский ГАУ»: СМК ПНД 77-01-2022, введено в действие приказом от 03.08.2015 №268а-О (<http://nsau.edu.ru/file/104821>: режим доступа свободный).