|  |  |
| --- | --- |
| Приложение 4 к рабочей программе дисциплины | |
| Алгоритмизация и программирование | |
| **Фонд оценочных средств** | |
| Направление/ специальность подготовки | 15.04.03 Прикладная механика |
| Специализация/ профиль/ программа подготовки | Методы и средства оценки напряженно-деформированного состояния, неразрушающего контроля, диагностики структуры и дефектности материалов в процессах пластического формоизменения |
| Уровень высшего образования | Магистратура |
| Форма обучения | Заочная |
| Факультет | Е Оружие и системы вооружения |
| Выпускающая кафедра | Е4 Высокоэнергетические устройства автоматических систем |
| Кафедра-разработчик | Е4 Высокоэнергетические устройства автоматических систем |
| Год приема | 2023 |

**ФОС по дисциплине «Алгоритмизация и программирование»**

**ОП ВО 15.04.03 Прикладная механика «Методы и средства оценки напряженно-деформированного состояния, неразрушающего контроля, диагностики структуры и дефектности материалов в процессах пластического формоизменения», форма обучения заочная**

ОПК-12. Способен создавать алгоритмы цифровой обработки баз данных результатов испытаний и эксплуатации сложных деталей и узлов в машиностроении, разрабатывать современные цифровые программы расчетов и проектирования деталей, узлов, конструкций, машин и материалов с учетом требований надежности, долговечности и безопасности их эксплуатации.

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер задания** | **Содержание вопроса** | **Компетенция** | **Время ответа, мин.** |
|  | Python является …  a. интерпретируемым языком.  b. компилируемым языком.  c. не интерпретируемым языком.  d. языком с интерпретацией компилирующего типа. | ОПК-12 | 1 |
|  | С точки зрения скорости выполнения программного кода, компилируемые языки программирования чаше всего …  a. медленнее интерпретируемых при использовании виртуальной машины.  b. быстрее интерпретируемых.  c. медленнее интерпретируемых.  d. аналогичны интерпретируемым. | ОПК-12 | 1 |
|  | Python является …  a. квазистатическим типизированным языком.  b. кинематически типизированным языком.  c. динамически типизированным языком.  d. статически типизированным языком. | ОПК-12 | 1 |
|  | При статической типизации …  a. переменная заранее не связывается с конкретным типом хранимых в ней данных, при этом в ней можно хранить сразу несколько разных типов.  b. переменная связывается с типом хранимых в ней данных в момент её объявления в программе, при этом её тип не может быть изменён позже.  c. переменная связывается с типом хранимых в ней данных в момент её в программе, при этом её тип может быть изменён позже.  d. переменная связывается с типом хранимых в ней данных в момент присвоения ей значения, при этом её тип может быть изменён позже при присвоении нового значения. | ОПК-12 | 3 |
|  | При динамической типизации …  a. переменная связывается с типом хранимых в ней данных в момент присвоения ей значения, при этом её тип может быть изменён при присвоении значении другого типа.  b. переменная связывается с типом хранимых в ней данных в момент её объявления в программе, при этом её тип не может быть изменён позже.  c. переменная связывается с типом хранимых в ней данных в момент её объявления в программе, при этом при необходимости изменения её типа придётся динамически создать новую переменную с требуемым типом.  d. переменная связывается с типом хранимых в ней данных в момент присвоения ей значения, при этом её тип не может быть изменён позже. | ОПК-12 | 3 |
|  | Императивное программирование отличается от декларативного тем, что …  a. при императивном стиле программирования реализуется объектно-ориентированная модель программирования, а при декларативном используется структурный стиль программирования.  b. при императивном стиле используется структурная парадигма программирования, а при декларативном реализуется объектно-ориентированная модель программирования.  c. при императивном стиле программирования описывается алгоритм получения желаемого результата, а при декларативном описывается какой именно результат должен быть получен в результате работы программы.  d. при императивном стиле программирования описывается какой именно результат должен быть получен в результате работы программы, а при декларативном алгоритм получения желаемого результата. | ОПК-12 | 3 |
|  | При структурном программировании …  a. программа представляется в виде иерархической структуры блоков.  b. программа создаётся путём манипуляции с графическими объектами.  c. программа представляется в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.  d. процесс вычисления трактуется как вычисление значений функций в математическом понимании последних. | ОПК-12 | 3 |
|  | При объектно-ориентированном программировании …  a. процесс вычисления трактуется как вычисление значений функций в математическом понимании последних.  b. программа представляется в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.  c. программа создаётся путём манипуляции с графическими объектами.  d. программа представляется в виде иерархической структуры блоков. | ОПК-12 | 3 |
|  | При функциональном программировании …  a. процесс вычисления трактуется как вычисление значений функций в математическом понимании последних.  b. программа представляется в виде иерархической структуры блоков  c. программа создаётся путём манипуляции с графическими объектами.  d. программа представляется в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования. | ОПК-12 | 3 |
|  | При визуальном программировании …  a. программа представляется в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.  b. процесс вычисления трактуется как вычисление значений функций в математическом понимании последних.  c. программа создаётся путём манипуляции с графическими объектами.  d. программа представляется в виде иерархической структуры блоков. | ОПК-12 | 3 |
|  | Рекурсивная функция …  a. это функция обладающая асимптотической сложность О(1).  b. это любая функция заданная в программном коде.  c. это функция, заданная самоподобным образом.  d. это функция задающая однозначные соответствия между переменными величинами на всём области её определения. | ОПК-12 | 3 |
|  | Вычислительная сложность …  a. это временные затраты на разработку алгоритма решения поставленной программисту задачи.  b. это когда процесс численных вычислений имеет стохастический характер.  c. это понятие, обозначающее функцию зависимости объема работы алгоритма от размера обрабатываемых данных.  d. это частный случай работы программы, когда алгоритм задан самоподобным образом. | ОПК-12 | 3 |
|  | Выберите наихудшую асимптотическую сложность работы алгоритма.  a. О(logN)  b. О(N)  c. О(N^2)  d. О(N·logN) | ОПК-12 | 3 |
|  | Выберите наихудшую асимптотическую сложность работы алгоритма.  a. О(N·logN)  b. О(1)  c. О(logN)  d. О(N) | ОПК-12 | 3 |
|  | Выберите наилучшую асимптотическую сложность работы алгоритма.  a. О(N·logN)  b. О(logN)  c. О(N)  d. О(1) | ОПК-12 | 3 |
|  | Выберите наилучшую асимптотическую сложность работы алгоритма.  a. О(N·logN)  b. О(N^2)  c. О(logN)  d. О(N) | ОПК-12 | 3 |
|  | Выберите верное утверждение.  a. Python является сугубо функциональным языком программирования.  b. Python является мультипарадигменным языком программирование.  c. Python является сугубо объектно-ориентированным языком программирования.  d. Python является сугубо структурным языком программирования. | ОПК-12 | 2 |
|  | Что выведет следующий код:  data = [1, 2, 5, 4, 1]  total = 0  for el in data:  if el % 2 == 0:  total += el  print(total) | ОПК-12 | 5 |
|  | Что выведет следующий код:  string = 'В теории, теория и практика неразделимы. На практике это не так.'  choosen\_word = string.split()[4]  length = len(choosen\_word)  print(length) | ОПК-12 | 5 |
|  | Что выведет следующий код:  counter = 1  value = 1  while counter < 5:  if value % 2 == 0:  value += 2  elif value % 2 == 1:  value += 1  break  counter += 1  print(value) | ОПК-12 | 5 |
|  | Что выведет следующий код:  counter = 1  value = 1  while counter < 5:  if value % 2 == 0:  value += 2  elif value % 2 == 1:  value += 1  pass  counter += 1  print(value) | ОПК-12 | 5 |
|  | Какое значение примет переменная:  var = '4' + '5'  a. 45  b. '45'  c. 9  d. '9'  e. Программа выдаст ошибку | ОПК-12 | 5 |
|  | Что выведет следующий код:  var = 1/3  print(f'{var:.3f}') | ОПК-12 | 5 |
|  | Что выведет следующий код:  total = 0  for i in range(2,9,3):  total += i  print(total) | ОПК-12 | 5 |
|  | Что выведет следующий код:  data = {'a': [3, 2], 'b': [4, 3], 'c': [1, 3, 2]}  ans = 0  for key in data.keys():  sub\_var = 1  for element in data[key]:  sub\_var \*= element  ans += sub\_var  print(ans) | ОПК-12 | 5 |
|  | Выберите неизменяемые типы данных   |  |  | | --- | --- | | Кортежи (tuple) |  | | Списки (list) |  | | Строки (str) |  | | Словари (dict) |  | | ОПК-12 | 5 |
|  | Что выведет следующий код:  string = 'штурм'  string[2] = 'о'  print(string)  a. 'шуурм'  b. 'шторм'  c. 'штурм'  d. 'о'  e. Исключение TypeError | ОПК-12 | 5 |
|  | Что выведет следующий код:  a = 1  b = a  b = 2  print(a) | ОПК-12 | 5 |
|  | Какая ошибка допущена в функции вычисления n-члена ряда Фибоначчи:  def fib\_calc(n):  return fib\_calc(n-1)  a. Отсутствует обработка базового случая рекурсии  b. Нельзя вызывать в функции саму себя  c. Не задано значение аргумента по-умолчанию  d. Отсутствует объявление типа переменной  e. Ошибка в синтаксисе задания функции | ОПК-12 | 5 |
|  | Как вывести справку по объекту из строк документации?  a. info(x)  b. dip(x)  c. help(x)  d. doc(x) | ОПК-12 | 2 |
|  | Пусть задан кортеж:  t = (1, 2, 3)  Как сделать чтобы программа возвращала кортеж типа:  (1, 2, 4)  a. t[:-1] + (4,)  b. t[3] = 4  c. t[:-1] + (4)  d. t[2] = 4 | ОПК-12 | 5 |
|  | Выберите, что нужно подставить вместо \_, чтобы программа вывела ответ:  5  Программа  sum = 0  for i in range(\_):  sum += 1  print(i) | ОПК-12 | 3 |
|  | Как подключить дополнительную библиотеку?  a. connect library  b. plug library  c. add library  d. #include library  e. import library | ОПК-12 | 1 |
|  | Как вычислить возведение 3 в степень 5?  a. 3\*\*5  b.  def func(num, p):  ans = 0  for i in range(num):  ans \*= i  return ans  func(3, 5)  c. mod(5, 3)  d. int(3)^5  e. 3^5 | ОПК-12 | 3 |
|  | Как добавить новую запись в словарь?  a. dict(key) = 'value'  b. dict[key] = 'value'  c. dict.append(value)  d. dict(key, 'value') | ОПК-12 | 2 |
|  | Выберите, что нужно подставить вместо \_, чтобы программа верно определила площадь окружности  import math  r = 3  area = \_  print(area)  a. math.pi \* r\*\*2  b. pi \* r ^ r  c. pi \* r\*\*2  d. math.pi \* r^2  e. math['pi'] \* r\*\*2 | ОПК-12 | 4 |
|  | Как получить матрицу вида:    a.  import numpy as np  arr = np.array((3, 3))  arr = np.numerate(1, 9, 1)  print(arr)  b.  import numpy as np  arr = np.linspace(1, 10)  arr = np.reshape(arr, (3, 3))  print(arr)  c.  import numpy as np  arr = np.arange(1, 10)  print(arr)  d.  import numpy as np  arr = np.arange(1, 10)  arr = np.reshape(arr, (3, 3))  print(arr)  e.  import numpy as np  arr = np.zeros((3, 3))  arr = np.numerate(1, 9, 1)  print(arr) | ОПК-12 | 4 |
|  | Выберите правильный синтаксис для создания массива из списка  a. numpy.linspace(1, 2, 3)  b. numpy.massive(1, 2, 3)  c. numpy.array([1, 2, 3])  d. numpy.massive([1, 2, 3])  e. numpy.linspace([1, 2, 3])  f. numpy.array(1, 2, 3) | ОПК-12 | 2 |
|  | Какой функцией можно создать массив 3х3, состоящий только из нулей?  a. numpy.null([3, 3])  b. numpy.zeros(3, 3)  c. numpy.zeros([3, 3])  d. numpy.empty(3, 3)  e. numpy.null(3, 3)  f. numpy.empty([3, 3]) | ОПК-12 | 2 |
|  | Как определить количество элементов в каждом измерении массива arr?  a. arr.size  b. arr.shape()  c. len(arr)  d. length(arr)  e. arr.shape  f. arr.size()  g. size | ОПК-12 | 4 |
|  | Какая из библиотек используется непосредственно для работы с матрицами?  a. scipy  b. sklearn  c. numpy  d. pandas  e. matplotlib | ОПК-12 | 1 |
|  | Что нужно подставить вместо \_\_\_\_, чтобы построить 2D график из линий?  import matplotlib.pyplot as plt  x = [1, 2, 3, 4, 5]  y = [1, 4, 9, 16, 25]  plt.\_\_\_\_(x, y)  a. plot  b. bar  c. scatter  d. hist  e. graph | ОПК-12 | 3 |
|  | Какая из библиотек используется для построения графиков?  a. pandas  b. matplotlib  c. scipy  d. sklearn  e. numpy | ОПК-12 | 1 |
|  | Что нужно подставить вместо \_\_\_\_, чтобы построить точечный график?  import matplotlib.pyplot as plt  x = [1, 2, 3, 4, 5]  y = [1, 4, 9, 16, 25]  plt.\_\_\_\_(x, y)  a. scatter  b. bar  c. plot  d. graph  e. hist | ОПК-12 | 2 |
|  | Что нужно подставить вместо \_\_\_\_, чтобы построить гистограмму?  import matplotlib.pyplot as plt  x = [1, 1, 3, 5, 4, 5, 6, 3, 3, 2, 1]  plt.\_\_\_\_(x)  a. hist  b. bar  c. plot  d. graph  e. scatter | ОПК-12 | 2 |
|  | Какой функцией сохранить график как изображение?  a. savefigure  b. printscreen  c. savefig  d. print  e. save | ОПК-12 | 2 |
|  | Какая из библиотек реализует множество математических алгоритмов, подходящих для инженерных и научных вычислений (интерполяция, интегрирование, решение уравнений и т. д.)?  a. pandas  b. sklearn  c. matplotlib  d. scipy  e. numpy | ОПК-12 | 1 |
|  | Какая из библиотек лучше всего подходит обработки и анализа табличных данных?  a. scipy  b. pandas  c. sklearn  d. numpy  e. matplotlib | ОПК-12 | 1 |
|  | Какая функция из scipy подойдёт для одномерной интерполяции зависимости y от x?  spl = interpolate.\_\_\_\_(x, y)  a. int  b. interpolation1d  c. interp1d  d. interpolation  e. interp | ОПК-12 | 3 |
|  | Как скалярно умножить матрицы arr1 и arr2 с помощью numpy?  a. arr1 \* arr2  b. numpy.scalar(arr1, arr2)  c. numpy.vector(arr1, arr2)  d. numpy.dot(arr1, arr2)  e. numpy.cross(arr1, arr2) | ОПК-12 | 2 |
|  | Как векторно умножить матрицы arr1 и arr2 с помощью numpy?  a. numpy.vector(arr1, arr2)  b. numpy.scalar(arr1, arr2)  c. arr1 \* arr2  d. numpy.dot(arr1, arr2)  e. numpy.cross(arr1, arr2) | ОПК-12 | 2 |
|  | Как поэлементно умножить элементы матриц arr1 и arr2 с помощью numpy?  a. numpy.cross(arr1, arr2)  b. numpy.scalar(arr1, arr2)  c. numpy.vector(arr1, arr2)  d. arr1 \* arr2  e. numpy.dot(arr1, arr2) | ОПК-12 | 2 |