Рабочая тетрадь № 7

|  |
| --- |
| Слово «Алгоритм» произошло от algorithmi – латинского написания имени аль-Хорезми, под которым в средневековой Европе знали величайшего математика из Хорезма (город в современном Узбекистане) Мухаммеда бен Мусу, жившего в 783-850 гг.  Как правило алгоритмы разрабатываются для конкретных исполнителей с их допустимыми действиями. Множество допустимых действий исполнителя определяет систему его команд (СКИ). В алгоритме должны содержаться только те действия, которые входят в систему команд для данного исполнителя. |

|  |
| --- |
| Основными свойствами алгоритмов выступают:  1. Массовость - применимость алгоритма к различным наборам исходных данных.  2. Дискретность - процесс решения задачи по алгоритму разбит на отдельные действия.  3. Однозначность - правила и порядок выполнения действий алгоритма имеют единственное толкование.  4. Конечность - каждое из действий и весь алгоритм в целом обязательно завершаются.  5. Результативность - по завершении выполнения алгоритма обязательно получается конечный результат.  6. Выполнимость - результата алгоритма достигается за конечное число шагов.  В литературе и на практике наиболее распространены формы описания алгоритмов представленные ниже:   * словесная (текст на естественном языке); * программная (программы на языках программирования); * графическая (изображения из графических символов); * псевдокоды (полуформализованные описания алгоритмов на условном алгоритмическом языке); |

|  |
| --- |
| **1. Теоретический материал** |
| ***Структурная (блок-, граф-) схема алгоритма*** – графическое представление алгоритма в виде связанных между собой с помощью линий перехода блоков – графических символов. Каждый блок соответствует одному шагу алгоритма (либо подпрограмме). Внутри блока приводится описание определенного действия.  Ключевые графические обозначения, используемые при создании блок-схем приведены в таблице: |

|  |  |
| --- | --- |
| **2. Пример** | |
| ***Задача:*** | |
|  | |  |  | | --- | --- | |  | Дан алгоритм в виде блок-схемы.  Найти А, В, С, D, если изначально:  А=10, В=10, C=4, D=0. | |
| ***Решение:*** | |
|  | Результат работы алгоритма определяется с помощью трассировочной таблицы (а, б, в, г): |
| ***Ответ:*** | |
|  | 10, 10, 10, 4 |
| ***Задача:*** | |
|  | Из ряда чисел 15, 16, 17, 18 выписать значения x, удовлетворяющие условию |
| ***Решение:*** | |
|  | Используя трассировочную таблицу, получим: |
| ***Ответ:*** | |
|  | 17, 18 |
| ***Задача:*** | |
|  | Дана блок-схема. Какое значение будет иметь N на выходе, если S=1.1? |
| ***Решение:*** | |
|  | Для определения емся трассировочной таблицей |
| ***Ответ:*** | |
|  | 2 |

|  |  |
| --- | --- |
| ***Задача:*** | |
|  | Дана блок-схема. Какое значение будет иметь z на выходе, если x=6? |
| ***Решение:*** | |
|  | Для определения результата воспользуемся трассировочной таблицей |
| ***Ответ:*** | |
|  | 2,875 |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | |  |  | | --- | --- | |  | Дан алгоритм в виде блок-схемы.  Найти А, В, С, D, если изначально:  а) А=0, В=0, C=5, D=10;  б) А=0, В=5, C=0, D=10;  в) А=10, В=20, C=6, D=4; | |
| ***Решение:*** | |
|  | а) А = 0, B = 0,C = 0, D = 5 Б) A = 0, B = 0, C = 5, D = 0 В) A = 10, B = 10, C = 20, D = 61 |
| ***Ответ:*** | |
|  | А) 0,0,0,5 б)0,0,5,0 В) 10,10,20,6 |
| 2. | ***Задача:*** | |
|  | Дана блок-схема. Какое значение будет иметь N на выходе, если S=2.09? |
| ***Решение:*** | |
|  | 1 шаг) s1 1, n 1, n = 2, s = 1,5 2) s1 = 1, n = 2, n = 3, s1 = 1,8 3) 1,8 3 4 2,08  4) 2,08 4 5 2,8 5) 2,8 > 2,09 конец, n = 5 |
| ***Ответ:*** | |
|  | 5 |

|  |  |  |
| --- | --- | --- |
| 3. | ***Задача:*** | |
|  | |  |  | | --- | --- | |  | Дана блок-схема.  Какое значение будет иметь z на выходе, если  а) x = 2,  б) х = 4? | |
| ***Решение:*** | |
|  | А) 1 шаг z = (1+2/2)/2 = 1, i 1. 2 шаг) i 2, z = 1 3) шаг i = 3, z = 1 4) шаг i = 4, z = 1 б) z = (1+4/2)/2 = 1,5, i = 1; z = (1,5+4/2)/2 = 1,75, i = 2; z = 1,875, i =3; z = 1,9375, i = 4 |
| ***Ответ:*** | |
|  | А) z = 1. Б) z = 1,9375 |

|  |
| --- |
| **1. Теоретический материал** |
| Машина Тьюринга – это абстрактный исполнитель, состоящий из бесконечной ленты, разделенной на ячейки, и автомата (головки), которая управляется программой. При этом ячейки могут содержать один символ из заданного алфавита A={a0, a1,…, aN}. Подразумевается, что любой алфавит включает пробельный символ (заменяется знаком подчеркивания «\_»). Данный абстрактный исполнитель управляется специальной таблицей, строки которой соответствуют символам алфавита A, а столбцы – определенным состояниям автомата Q ={q0,q1,…, qM}. Перед началом работы исполнитель находится в состоянии q1, а состояние q0 – это конечное, при котором машина Тьюринга останавливается.  В каждой клетке таблицы, соответствующей некоторому символу ai и некоторому состоянию qj, находится команда, состоящая из трех частей:   1. символ алфавита; 2. направление перемещения головки: влево, вправо или остаться на месте; 3. новое состояние автомата   Внешний вид интерпретатора представлен на рисунке:  http://kpolyakov.spb.ru/prog/images/turwin.gif  Тренажер и инструкцию к программе для изучения универсального исполнителя «Машина Тьюринга» можно найти по ссылке (также лежит на сетевом диске «.\Сетевой диск\Информатика\_2020\Рабочая тетрадь 7»)  <http://kpolyakov.spb.ru/prog/turing.htm> |

|  |  |  |
| --- | --- | --- |
| **2. Пример** | | |
| 1. | ***Задача:*** | |
|  | Написать программу на машине Тьюринга, прибавляющую число 2 к введенному числу. |
| ***Решение:*** | |
|  | I:\Информатика_2020\Рабочие тетради\рт7\3.png |
| 2. | ***Задача:*** | |
|  | Удалить из слова его второй символ, если такой есть. Алфавит: A={a, b}. |
| ***Решение:*** | |
|  | Необходимо запомнить первый символ, стереть его на ленте, перейти направо и вместо второго символа записать первый.  I:\Информатика_2020\Рабочие тетради\рт7\2.png |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Написать на машине Тьюринга программу, прибавляющую 5 к введенному числу. |
| ***Решение:*** | |
|  |  |
| 2. | ***Задача:*** | |
|  | Если **P** – непустое слово, то за его первым символом вставить символ **a**. Алфавит: A={a, b, c}. |
| ***Решение:*** | |
|  |  |
| 3. | ***Задача:*** | |
|  | Вставить в слово **P** символ **a** за первым символом **c**, если такое есть. Алфавит: A={a, b, c}. |
| ***Решение:*** | |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Тест 7** | | |
| **1.** | Алгоритм это: | |
|  | а) аналог, образ какого-либо объекта, процесса или явления, сохраняющий его существенные черты  б) пошаговое описание последовательности действий, которые необходимо, выполнить для решения задачи  в) описание существенных для поставленной задачи свойств и закономерностей поведения объектов, обеспечивающее её решение  г) программа, предназначенная для создания и обработки графической информации; |
| ***Ответ:*** | |
|  |  |
| **2.** | Свойство алгоритма, которое обеспечивает решение целого класса задач одного типа: | |
|  | а) понятность; б) определенность;  в) дискретность; г) массовость. |
| ***Ответ:*** | |
|  |  |
| **3.** | Как называется конструкция блок-схемы, изображенная на рисунке? | |
|  | а) вызов вспомогательного алгоритма  б) начало-конец алгоритма  в) выполнение операций  г) ввод/вывод данных |
| ***Ответ:*** | |
|  |  |
| **4.** | На рисунке представлена часть блок-схемы. Как она называется? | |
|  | а) альтернатива;  б) композиция;  в) цикл с предусловием;  г) итерация? |
| ***Ответ:*** | |
|  |  |
| **5.** | Определите значения переменной «x» после выполнения фрагмента алгоритма. | |
|  | https://kpolyakov.spb.ru/school/test7a/16_files/9.gif |
| ***Ответ:*** | |
|  | 3 |
| **6.** | Определите значение переменной «b» после выполнения фрагмента алгоритма. | |
|  | https://kpolyakov.spb.ru/school/test7a/16_files/10.gif |
| ***Ответ:*** | |
|  | 25 |
| **7.** | В машине Тьюринга предписание **S** для лентопротяжного механизма означает | |
|  | а) переместить ленту вправо;  б) переместить ленту влево;  в) остановить машину  г) занести в ячейку символ |
| ***Ответ:*** | |
|  |  |
| **8.** | Конечный автомат – это автомат с …. | |
|  | а) конечным числом выходов  б) конечным числом состояний  в) конечным числом входов  г) конечным числом функций |
| ***Ответ:*** | |
|  |  |
| **9.** | Пусть дана машина Тьюринга с внешним алфавитом ∑ = {a,b, \_} и состояниями Q = {q0, q1, q2}. Программа приведена в таблице:    В какое слово программа преобразует слово *abb*, если головка находится над первой буквой*?* | |
|  | а) aab б) abbb в) abba г) aabb |
| ***Ответ:*** | |
|  |  |
| **10.** | Машина Тьюринга задана программой, записанной в виде таблицы:    Определите, в какое слово машина Тьюринга преобразует слово «111», если головка находится над первой (слева направо) единицей? | |
|  | а) 111 б) 0  в) 001 г) машина не останавливается |
| ***Ответ:*** | |
|  |  |

Реализация задач на языке программирования Python

|  |
| --- |
| При написании коль сколько-нибудь сложных программ необходимо код программы разбивать на отдельные части. Самым мелкой частью являются функции, которые программируются автономно от всей программы, но которые связаны с помощью входных параметров, и возвращаемым результатом. |

|  |
| --- |
| **1. Теоретический материал** |
| Сначала рассмотрим определение и использование функции в языке Python. Как мы уже отмечали, функции могут иметь параметры и возвращать значения. При этом операторы тела функции будут активированы в момент вызова функции из другой части кода. Вот пример функции:  # определение функции  def func(a, b):  c = a \* b  return c  # основная часть программы  n = 5  m = 6  k = func(n, m)  print(k)  При создании функции в Python не нужно определять ни тип входных параметров, ни тип результата функции. Однако при определении функции можно указать значение аргументов по умолчанию:  # определение функции  def func(a, b = 3):  c = a \* b  return c  # основная часть программы  n = 5  k = func(n)  print(k)  Функции можно вызывать из других функций, в частности рекурсивно, то есть из самой функции. Приведем пример рекурсии для вычисления факториала:  def fact(n):  ifn<= 1:  return 1  else:  return n \* fact(n - 1)    N = int(input("Введите N > "))  NF = fact(N)  print("N! = " + NF)  С помощью конструкции кортежей можно возвращать более одного значения функции. Рассмотрим пример функции, которая возвращает одновременно минимальное и максимальное значения.  def MinMax(P):  if len(P) < 1:  return (0, 0)  Min = P[0]  Max = P[0]  for x in P:  if x < Min:  Min = x  if x > Max:  Max = x  return (Min, Max)    P = [-1, 10, 2, 3, -2, 5, 6]  Min, Max = MinMax(P)  print("Min = " + Min.\_\_str\_\_() + ", Max = " + Max.\_\_str\_\_()) |

|  |  |
| --- | --- |
| **2. Пример** | |
| ***Задача:*** | |
|  | При написании программ существует возможность вызова одной функции из другой. Приведем пример реализации этой возможности. Основная ветка программы должна содержать одну инструкцию – вызов функции **test()**. В функции **test()** требуется запросить ввод целого числа. Если введенное число положительное, необходимо вызвать функцию **positive()**, тело которой содержит команду вывода в консоль слова "Positive". В противном случае требуется вызвать функцию **negative()**, которая выводит в консоль слово "Negative". |
| ***Решение (код программы):*** | |
|  | def positive():  print("Positive")  def negative():  print("Negative")  def test(x):  if x > 0:  positive()  else:  negative()  test(-5) |
| ***Задача:*** | |
|  | Напишите программу, в которой вызывается функция, в качестве аргументов принимающая две строки и возвращающая в программу результат их конкатенации (слияния). Выведите результат на экран. |
| ***Решение (код программы):*** | |
|  | def sum(a, b):  return a + b  print(sum("asd", "dsa")) |
| ***Задача:*** | |
|  | В зависимости от выбора пользователя вычислить площадь круга, прямоугольника или треугольника. Для вычисления площади каждой фигуры должна быть написана отдельная функция. |
| ***Решение (код программы):*** | |
|  | import math  def circle(r):  return math.pi \* r\*\*2  def rectangle(a, b):  return a\*b  def triangle(a, b, c):  p = (a+b+c)/2  return math.sqrt(p \* (p-a) \* (p-b) \* (p-c))  choice = input("Круг(к), прямоугольник(п) или треугольник(т): ")  if choice == 'к':  rad = float(input("Радиус: "))  print("Площадь круга: %.2f" % circle(rad))  elif choice == 'п':  l = float(input("Длина: "))  w = float(input("Ширина: "))  print("Площадь прямоугольника: %.2f" % rectangle(l,w))  elif choice == 'т':\  AB = float(input("Первая сторона: "))  BC = float(input("Вторая сторона: "))  CA = float(input("Третья сторона: "))  print("Площадь треугольника: %.2f" % triangle(AB,BC,CA)) |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Напишите функцию, которая на вход принимает два целых числа (делимое и делитель) и возвращает целую часть от деления и остаток. |
| ***Решение (код программы):*** | |
|  | **def func(a,b):**  **c = a // b**  **return c**  **def func1(a,b):**  **d = a % b**  **return d**  **# основная часть программы**  **n = 10**  **b = 2**  **k = func(n,b)**  **l = func1(n,b)**  **print("Остаток",k," Целая часть от деления:" ,l)** |
| 2. | ***Задача:*** | |
|  | Описать функцию Sign(X) целого типа, возвращающую для вещественного числа X следующие значения: −1, если X < 0; 0, если X = 0; 1, если X > 0. |
| ***Решение (код программы):*** | |
|  | def Sign(X):  if X < 0:  return -1  elif X > 0:  return 1  return 0 |
| 3. | ***Задача:*** | |
|  | Напишите проверку на то, является ли строка палиндромом. Палиндром — это слово или фраза, которые одинаково читаются слева направо и справа налево. |
| ***Решение (код программы):*** | |
|  | slovo = str(input())  a = slovo[::-1]  if slovo == a:  print("yes")  else:  print("no") |
| 4. | ***Задача:*** | |
|  | Сделайте так, чтобы число секунд отображалось в видедни:часы:минуты:секунды. |
| ***Решение (код программы):*** | |
|  | sec = int(input())    day = 86400  hour = 3600  minute = 60  second = 1    days = sec // day  hours = (sec % day) // hour  minutes = ((sec % day) % hour) // minute  seconds = ((sec % day) % hour) % minute  print(days, hours, minutes, seconds, sep=':') |
| 5. | ***Задача:*** | |
|  | Напишите функцию, которая вычисляет любое число Фибоначчи с помощью рекурсии. |
| ***Решение (код программы):*** | |
|  | **fib1 = 0**  **fib2 = 1**    **n = int(input("Номер элемента ряда Фибоначчи: "))**    **i = 0**  **while i < n - 2:**  **fib\_sum = fib1 + fib2**  **fib1 = fib2**  **fib2 = fib\_sum**  **i = i + 1**    **print("Значение этого элемента:", fib2)** |