

Темы занятий для начинающих

Занятие 1. Рассмотрено понятие алгоритма, перечислены основные свойства алгоритмов, а также способы их записи. Рассказано про основные виды алгоритмов: линейные, с ветвлением, циклические, рекурсивные, вспомогательные. Введено понятие переменной. На примере языке Python рассказано про оператор присваивания и основные типы данных в Python. На занятии рассказывалось про целые и вещественные числа, операции над ними. Особое внимание уделялось операциям деления с округлением вниз (`//`) и делению с остатком (`%`). Рассмотрен строковый тип и операции над строками, а также логический тип данных. Для самостоятельного решения были предложены задачи на целочисленную арифметику.

Занятие 2. Рассмотрены задачи на разбор случаев. Рассказано про условный оператор в Python: синтаксис, вложенный условный оператор, каскадный условный оператор. Разобраны примеры задач, требующих использование условного оператора. Подробно рассмотрен логический тип данных, логические операции и операции сравнения. Для самостоятельного решения были предложены задачи на разбор случаев.

Занятие 3. На занятии рассматривались задачи требующие повторения одних и тех же действий некоторое количество раз. Введено понятие цикла с предусловием и постусловием. На примере языка Python было рассказано про цикл с параметром `for`: синтаксис, особенности использования, функцию `range`. Также были разобраны задачи на обработку числовых последовательностей, например поиск максимального элемента за один просмотр чисел. Для самостоятельного решения были предложены задачи на циклы с известным числом повторений.

Занятие 4. Рассмотрен синтаксис цикла `while`. Рассказаны основные ошибки, допускаемые при написании циклов и приводящие к заиклииванию. Рассмотрены операторы `break` (для досрочного выхода из цикла) и `continue` (для досрочного перехода к следующей итерации цикла). Также на примерах было показано использование инструкции `else` для циклов в Python. Ещё одна затронутая на занятии тема – сложность алгоритмов и важность написания эффективных алгоритмов для решения задач. В качестве примеров задач рассматривались задачи на обработку последовательностей, а также алгоритм эффективного нахождения делителей числа. Для самостоятельного решения были предложены задачи на циклы с условием.

Занятие 5. Контрольная работа по пройденным темам. Было предложено по одной задаче на целочисленную арифметику, условный оператор, цикл `for`, цикл `while`, вложенные циклы. После завершения контрольной работы был проведён разбор задач.

Примеры задач для начинающих

1. Число 179 записали 50 раз подряд. Полученное 150-значное число возвели в квадрат. Сколько получилось?

2. В некоторой школе решили набрать три новых математических класса и оборудовать кабинеты для них новыми партами. За каждой партой может сидеть два учащихся. Известно количество учащихся в каждом из трех классов. Выведите наименьшее число парт, которое нужно приобрести для них.
3. Даны три натуральных числа a , b , c . Определите, существует ли треугольник с такими сторонами. Если треугольник существует, выведите строку YES, иначе выведите строку NO.
4. Есть две коробки, первая размером $A1 \times B1 \times C1$, вторая размером $A2 \times B2 \times C2$. Определите, можно ли разместить одну из этих коробок внутри другой, при условии, что поворачивать коробки можно только на 90 градусов вокруг ребер.
5. Дано несколько чисел. Подсчитайте, сколько из них равны нулю, и выведите это количество.
6. По данному натуральному $n \leq 9$ выведите лесенку из n ступенек, i -я ступенька состоит из чисел от 1 до i без пробелов.
7. Дано натуральное число N . Выведите слово YES, если число N является точной степенью двойки, или слово NO в противном случае. Операцией возведения в степень пользоваться нельзя!
8. Последовательность состоит из натуральных чисел и завершается числом 0 . Определите, сколько элементов этой последовательности больше предыдущего элемента.

Темы занятий для продолжающих

Занятие 1. На занятии предлагалось решить 10 разных задач на разные синтаксические конструкции языка программирования Python для знакомства с тестирующей системой: задачи на ввод вывод данных, целочисленную арифметику, условный оператор, циклы `for` и `while`, списки. По каждой задаче проводился разбор основной идеи, используемой для решения задач.

Занятие 2. На занятии рассматривались встроенные и пользовательские функции. Было рассказано про способы подключения модулей в Python, синтаксисе пользовательских функций. Рассмотрено понятие рекурсивной функции, прямой и косвенной рекурсии. Особое внимание уделялось ошибкам, допускаемым при написании рекурсивных функций. Для самостоятельного решения были предложены задачи на написание собственных функций, в том числе рекурсивных.

Занятие 3. Разобраны наиболее сложные задачи на рекурсию из задач для самостоятельного решения с занятия 2. Затронут вопрос о неэффективности рекурсивных программ. Рассмотрены примеры использования рекурсии для генерации комбинаторных объектов и алгоритм перебора с возвратом, метод ветвей и метод границ. Для самостоятельного решения были предложены задачи на алгоритм перебора с возвратом.

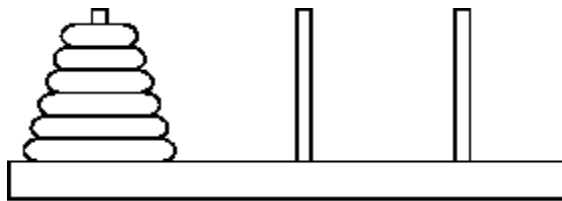
Занятие 4. Затронут вопрос о сложности алгоритмов и важности написания эффективных программ. Рассмотрен метод декомпозиции задач. На примерах рассказано про динамическое программирование: что это за способ решения задач, для каких задач он

подходит, как решать каждый тип задачи. Для самостоятельного решения были предложены задачи на одномерное и двумерное динамическое программирование.

Занятие 5. Контрольная работа по пройденным темам. Было предложено 5 задач, для каждой из которых требовалось выбрать наилучший из изученных методов решения. После завершения контрольной работы был проведён разбор задач.

Примеры задач для продолжающих

1. В списке все элементы различны. Поменяйте местами минимальный и максимальный элемент этого списка.
2. Решить в целых числах уравнение $(ax + b) : (cx + d) = 0$.
3. Напишите функцию быстрого возведения в степень. Количество действий должно быть пропорционально двоичному логарифму n .
4. Головоломка “Ханойские башни” состоит из трех стержней, пронумерованных числами 1, 2, 3. На стержень 1 надета пирамидка из n дисков различного диаметра в порядке возрастания диаметра. Диски можно перекладывать с одного стержня на другой по одному, при этом диск нельзя класть на диск меньшего диаметра. Необходимо переложить всю пирамидку со стержня 1 на стержень 3 за минимальное число перекладываний.



Напишите программу, которая решает головоломку; для данного числа дисков n печатает последовательность перекладываний в формате $a b c$, где a — номер перекладываемого диска, b — номер стержня с которого снимается данный диск, c — номер стержня на который надевается данный диск.

1. По данным числам N и K выведите все строки длины N из символов $0..K-1$ в лексикографическом порядке.
2. По данным числам N и K выведите все возрастающие последовательности длины K из чисел $1..N$ в лексикографическом порядке.
3. При переработке радиоактивных материалов образуются отходы двух видов — особо опасные (тип А) и неопасные (тип В). Для их хранения используются одинаковые контейнеры. После помещения отходов в контейнеры последние укладываются вертикальной стопкой. Стопка считается взрывоопасной, если в ней подряд идет более одного контейнера типа А. Стопка считается безопасной, если она не является взрывоопасной. Для заданного количества контейнеров N определить количество возможных типов безопасных стопок.
4. В каждой клетке прямоугольной таблицы $N \times M$ записано некоторое число. Изначально игрок находится в левой верхней клетке. За один ход ему разрешается перемещаться в соседнюю клетку либо вправо, либо вниз (влево и вверх перемещаться запрещено). При проходе через клетку с игрока берут столько килограммов еды, какое число записано в этой клетке (еду берут также за первую и последнюю клетки его пути). Требуется найти

минимальный вес еды в килограммах, отдав которую игрок может попасть в правый нижний угол.