

Занятие 2.1. Упражнения

- 1) Пересмотрите функции, написанные вами на предыдущем занятии. Добейтесь более короткой записи с помощью задания списков диапазонами значений, выборки элементов списка, композиции, каррирования и частичного применения функций там, где это возможно.
- 2) Определите функцию, возвращающую последовательность n чисел Фибоначчи, начиная с m -го. По возможности используйте функции, определённые вами ранее.
- 3) Определите функцию, возвращающую последовательность n простых чисел, начиная с m -го. По возможности используйте функции, определённые вами ранее.
- 4) Определите функцию, возвращающую последовательность чисел в порядке возрастания в интервале $[a, b]$ с приращением d (аналог оператора `range` в языке Python): (а) для целых чисел, (б) для вещественных чисел.
- 5) Определите функцию, возвращающую подсписок, включающий n элементов исходного списка, начиная с элемента с индексом m (аналог функции `Mid` в Basic). Какой порядок аргументов такой функции вы бы выбрали?
- 6) Напишите свой вариант функции быстрой сортировки. Пусть ваша функция принимает в качестве одного из аргументов функцию, с помощью которой задаётся правило сравнения элементов (аналогично `qsort` в C). Продемонстрируйте работу функции на примерах: (а) сортировки слов по их длине, (б) сортировки только по первой и третьей буквам слова без учёта регистра этих букв.
- 7) Модифицируйте функцию поиска минимума функции одной переменной (предыдущее занятие) таким образом, чтобы: (а) глубина рекурсии была ограничена, (б) на каждой итерации в консоль выводились бы интервалы, на которых выполняется поиск минимума.
- 8) Реализуйте расчёт квартилей (верхней, нижней и средней — медианы) для произвольной выборки, представленной списком значений. Реализуйте функции расчёта интерквартильного размаха, максимума и минимума выборки (как исключения из исходного списка и включения в отдельные списки «выбросов» и «грубых выбросов» (значений, отстоящих от максимального и минимального значений выборки на 1,5 и 3 интерквартильных размахов соответственно). Реализацию оформите в виде отдельного модуля, подготовьте примеры.
- 9) Используя результаты предыдущего задания, реализуйте построение графика типа «ящик с усами». Функция должна принимать список значений и создавать файл в формате `LaTeX` и/или `SVG` с заданным именем. Пусть «выбросы» и «грубые выбросы» будут показаны на графике маркерами разного начертания. Обеспечьте возможность начертить два таких графика рядом для сравнения. Оформите в виде отдельного модуля, подготовьте примеры. В `LaTeX` высокоуровневые пакеты графики не используйте, используйте окружение `picture`.
- 10) Реализуйте построение графика произвольной функции на заданном интервале. Ваша функция должна принимать функцию, график которой следует построить, и список интервалов и создавать файл в формате `LaTeX` и/или `SVG` с заданным именем. В `LaTeX` высокоуровневые пакеты графики не используйте, используйте окружение `picture`. При необходимости сглаживания используйте кривые Безье. Оформите в виде отдельного модуля, подготовьте примеры, включающие в том числе тригонометрические функции и функции вида $y = 1/(kx)$, $y = 1/(kx^n)$.
- 11) Реализуйте собственные функции для проверки, является ли список множеством, преобразования списка к множеству и для выполнения всех основных операций над множествами. Оформите в виде отдельного модуля, подготовьте примеры.
- 12) Определите функцию, осуществляющую поиск в списке слов всех возможных словоформ заданного слова. Проверьте работоспособность функции на примерах слов из русского и английского языков. Рекомендация: для корректного вывода кириллицы в консоль используйте функции `putStr`, `putStrLn`.
- 13) Напишите программу, составляющую частотный словарь для произвольного текста. Проверьте работоспособность программы на примерах русских и английских текстов.
- 14) Модифицируйте программу из предыдущего задания таким образом, чтобы она по возможности учитывала встречаемость слов в различных словоформах. Проверьте работоспособность программы на примерах русских и английских текстов.
- 15) Реализуйте библиотеку функций для операций над векторами произвольной размерности. Оформите в виде отдельного модуля, подготовьте примеры.

Занятие 2.1. Упражнения

- 16) Реализуйте библиотеку функций для операций над матрицами. Оформите в виде отдельного модуля, подготовьте примеры.
- 17) Используя функции предыдущих заданий, реализуйте библиотеку функций для аффинных преобразований набора точек на плоскости и в трёхмерном пространстве: перенос, поворот вокруг произвольной точки на эйлеровы углы, масштабирование, отражение, получение перспективной проекции и т.д. Предусмотрите примеры, на которых можно показать корректность функций, не прибегая к графическим построениям, и примеры совместного использования этих функций.
- 18) Напишите функции, позволяющие работать со списком как со стеком, очередью, двусвязной очередью, кольцевым буфером. Оформите в виде отдельного модуля, подготовьте примеры.
- 19) Используя результаты предыдущего задания, напишите программу — консольный калькулятор. Пользователь должен вводить числа и операции через пробел после приглашения программы. Предусмотрите выход из программы по команде. Постарайтесь реализовать возможность ввода нескольких операций в строку (подразумевается, что используется обратная польская запись).
- 20) Реализуйте библиотеку интервальной арифметики. Обеспечьте возможность задавать интервальные числа по крайней мере двумя способами. Оформите в виде отдельного модуля. Проверьте работу модуля на примере задачи о параллельном соединении двух резисторов, рассчитав общее сопротивление по формулам $R = R_1 R_2 / (R_1 + R_2)$ и $R = 1 / (1/R_1 + 1/R_2)$. Совпадают ли результаты вычислений по этим формулам? Если не совпадают, предложите способ устранения этого недостатка.

Результаты работы сдайте преподавателю. Должны быть предоставлены исходные файлы и файлы примеров. В комментариях в каждом исходном файле укажите автора и кодировку. В комментариях к каждой функции укажите её назначение, что она принимает в качестве аргументов, приведите краткий пример её тестирования в консоли. Файлы примеров опишите в файле `readme.txt` (в начале этого файла укажите его кодировку!), для текстовых файлов также укажите кодировку, тип концов строк и признака конца файла (Windows или *nix). Теоретические справки с математическими формулами сдавать только в формате PDF. Результаты выполнения заданий сохраните для дальнейшей работы.