Задание № 1: Создание функции, возвращающей сумму двух чисел.

Напишите функцию *sum*, которая принимает два числа и возвращает их сумму.

Описание алгоритма решения:

1. Объявите функцию *sum* с двумя параметрами *a и b*.
2. Внутри функции верните результат сложения *a и b*

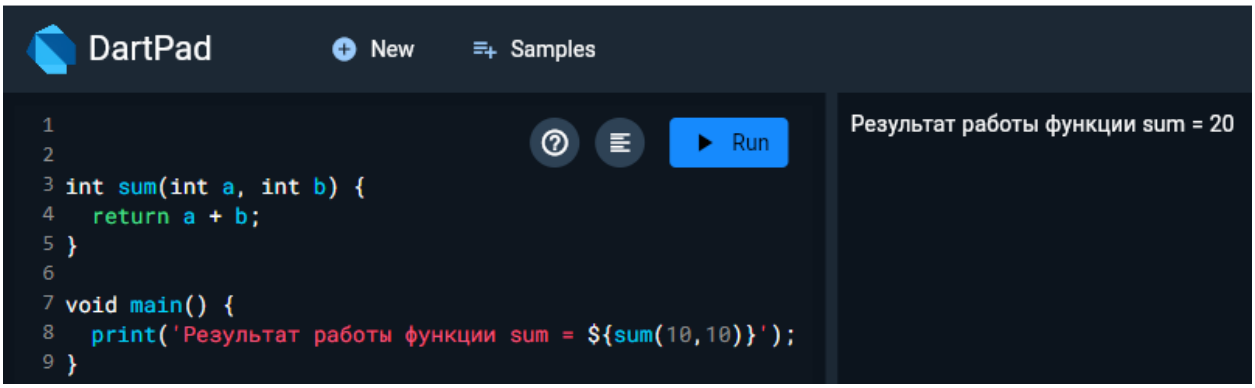

Рисунок №1 – Результат решения задания №1.

Задание № 2: Создание функции, возвращающей произведение двух чисел.

Напишите функцию *multiply*, которая принимает два числа и возвращает их произведение.

Описание алгоритма решения:

1. Объявите функцию *multiply* с двумя параметрами *a и b*.
2. Внутри функции верните результат умножения *a и b*.

Рисунок №2 – Результат решения задания №2.

Задание № 3: Создание функции, возвращающей результат деления двух чисел

Напишите функцию *divide*, которая принимает два числа и возвращает их частное.

Описание алгоритма решения:

1. Объявите функцию *divide* с двумя параметрами *a и b*.
2. Внутри функции верните результат деления *a на b*.

Рисунок №3 – Результат решения задания №3.

Задание № 4: Создание функции, возвращающей остаток от деления двух чисел

Напишите функцию *remainder*, которая принимает два числа и возвращает остаток от их деления.

Описание алгоритма решения:

1. Объявите функцию *remainder* с двумя параметрами *a и b*.
2. Внутри функции верните результат операции *a % b*.

Рисунок №4 – Результат решения задания №4.

Задание № 5: Создание функции, возвращающей результат возведения числа в степень

Напишите функцию *power*, которая принимает два числа и возвращает результат возведения первого числа в степень второго.

Описание алгоритма решения:

1. Объявите функцию *power* с двумя параметрами *base и exponent*.
2. Внутри функции верните результат возведения *base* в степень *exponent*.

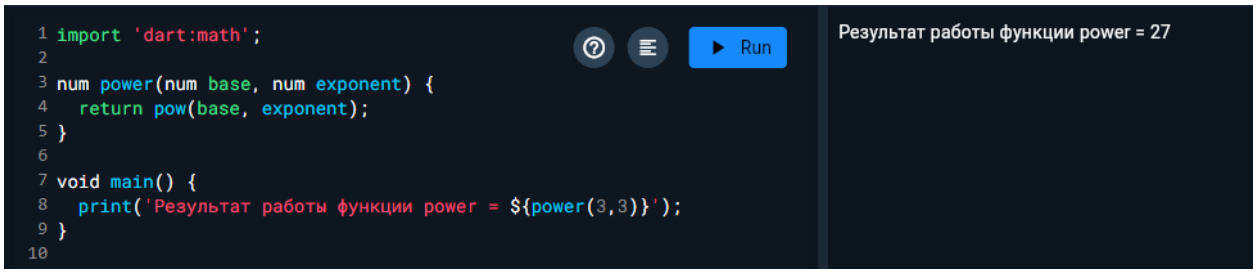

Рисунок №5 – Результат решения задания №5.

Задание № 6: Создание функции, возвращающей факториал числа

Напишите функцию *factorial*, которая принимает число и возвращает его факториал.

Описание алгоритма решения:

1. Объявите функцию *factorial* с одним параметром *n*.
2. Используйте цикл *for* для вычисления факториала числа *n*.
3. Верните результат.

Рисунок №6 – Результат решения задания №6.

Задание № 7: Создание функции, возвращающей сумму элементов списка

Напишите функцию *sumList*, которая принимает список чисел и возвращает их сумму.

Описание алгоритма решения:

1. Объявите функцию *sumList* с одним параметром *list*.
2. Используйте цикл *for* для перебора элементов списка и вычисления их суммы.
3. Верните результат.

Рисунок №7 – Результат решения задания №7.

Задание № 8: Создание функции, возвращающей среднее значение элементов списка

Напишите функцию *averageList*, которая принимает список чисел и возвращает их среднее значение.

Описание алгоритма решения:

1. Объявите функцию *averageList* с одним параметром *list*.
2. Используйте цикл for для перебора элементов списка и вычисления их суммы.
3. Разделите сумму на количество элементов в списке.
4. Верните результат.

Рисунок №8 – Результат решения задания №8.

Задание № 9: Создание функции, возвращающей максимальный элемент списка

Напишите функцию *maxElement*, которая принимает список чисел и возвращает максимальный элемент.

Описание алгоритма решения:

1. Объявите функцию *maxElement* с одним параметром *list*.
2. Используйте цикл *for* для перебора элементов списка и поиска максимального значения.
3. Верните результат.

Рисунок №9 – Результат решения задания №9.

Задание № 10: Создание функции, возвращающей минимальный элемент списка

Напишите функцию *minElement*, которая принимает список чисел и возвращает минимальный элемент.

Описание алгоритма решения:

1. Объявите функцию *minElement* с одним параметром *list*.
2. Используйте цикл *for* для перебора элементов списка и поиска минимального значения.
3. Верните результат.

Рисунок №10 – Результат решения задания №10.

Задание № 11: Создание функции, возвращающей список квадратов чисел

Напишите функцию *squareList*, которая принимает список чисел и возвращает список их квадратов.

Описание алгоритма решения:

1. Объявите функцию squareList с одним параметром list.
2. Используйте цикл for для перебора элементов списка и вычисления их квадратов.
3. Верните новый список с квадратами.

Рисунок №11 – Результат решения задания №11.

Задание № 12: Создание функции, возвращающей список четных чисел

Напишите функцию *evenNumbers*, которая принимает список чисел и возвращает список только четных чисел.

Описание алгоритма решения:

1. Объявите функцию *evenNumbers* с одним параметром *list*.
2. Используйте цикл *for* для перебора элементов списка и добавления четных чисел в новый список.
3. Верните новый список.

Рисунок №12 – Результат решения задания №12.

Задание № 13: Создание функции, возвращающей список нечетных чисел

Напишите функцию *oddNumbers*, которая принимает список чисел и возвращает список только нечетных чисел.

Описание алгоритма решения:

1. Объявите функцию *oddNumbers* с одним параметром *list*.
2. Используйте цикл *for* для перебора элементов списка и добавления нечетных чисел в новый список.
3. Верните новый список.

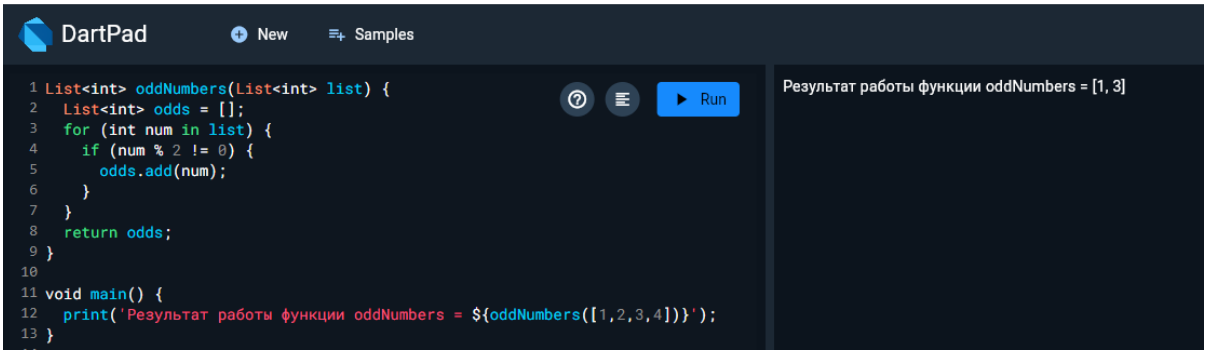

Рисунок №13 – Результат решения задания №13.

Задание № 14: Создание функции, возвращающей список чисел, больших заданного

Напишите функцию *greaterThan*, которая принимает список чисел и число *threshold*, и возвращает список чисел, больших *threshold*.

Описание алгоритма решения:

1. Объявите функцию *greaterThan* с двумя параметрами *list* и *threshold*.
2. Используйте цикл *for* для перебора элементов списка и добавления чисел, больших *threshold*, в новый список.
3. Верните новый список.

Рисунок №14 – Результат решения задания №14.

Задание № 15: Создание функции, возвращающей список чисел, меньших заданного

Напишите функцию *lessThan*, которая принимает список чисел и число *threshold*, и возвращает список чисел, меньших *threshold*.

Описание алгоритма решения:

1. Объявите функцию *lessThan* с двумя параметрами *list* и *threshold*.
2. Используйте цикл *for* для перебора элементов списка и добавления чисел, меньших *threshold*, в новый список.
3. Верните новый список.

Рисунок №15 – Результат решения задания №15.

Задание № 16: Создание функции, возвращающей список чисел, равных заданному

Напишите функцию *equalTo*, которая принимает список чисел и число *target*, и возвращает список чисел, равных *target*.

Описание алгоритма решения:

1. Объявите функцию *equalTo* с двумя параметрами *list* и *target*.
2. Используйте цикл *for* для перебора элементов списка и добавления чисел, равных *target*, в новый список.
3. Верните новый список.

Рисунок №16 – Результат решения задания №16.

Индивидуальные задания для закрепления материала

Базовые задания:

1. **Калькулятор**: Разработайте программу, которая выполняет базовые арифметические операции (сложение, вычитание, умножение, деление) над двумя числами. Используйте функции для каждой операции.
2. **Конвертер температур**: Создайте программу, которая конвертирует температуру между градусами Цельсия, Фаренгейта и Кельвина. Используйте отдельные функции для каждого преобразования.
3. **Генератор случайных чисел**: Напишите программу, которая генерирует случайное число в заданном диапазоне. Используйте функцию rand() и реализуйте функцию для задания диапазона.
4. **Проверка на простоту**: Разработайте функцию, которая проверяет, является ли заданное число простым. Используйте эту функцию в программе, которая находит все простые числа в заданном диапазоне.
5. **Сортировка массива**: Напишите программу, которая сортирует массив чисел по возрастанию. Используйте функцию для реализации алгоритма сортировки (например, пузырьковая сортировка).
6. **Поиск максимального элемента**: Разработайте функцию, которая находит максимальный элемент в массиве. Используйте эту функцию в программе, которая находит максимальный элемент в двумерном массиве.
7. **Работа со строками**: Напишите программу, которая подсчитывает количество символов, слов и строк в тексте. Используйте функции для каждой операции.
8. **Калькулятор площади фигур**: Разработайте программу, которая вычисляет площадь различных геометрических фигур (круг, прямоугольник, треугольник). Используйте отдельные функции для каждой фигуры.
9. **Игра «Угадай число»**: Напишите программу, в которой компьютер загадывает число, а пользователь пытается его угадать. Используйте функции для генерации случайного числа и проверки предположения пользователя.

Средний уровень:

1. **Система управления задачами**: Создайте консольное приложение для управления задачами. Пользователь может добавлять, просматривать, редактировать и удалять задачи. Используйте функции для каждого действия.
2. **Телефонный справочник**: Разработайте программу для хранения и управления телефонным справочником. Пользователь может добавлять, искать, редактировать и удалять записи. Используйте функции для каждой операции.
3. **Программа для шифрования текста**: Напишите программу, которая шифрует и дешифрует текст с использованием шифра Цезаря. Используйте функции для шифрования и дешифрования.
4. **Система управления банковским счетом**: Разработайте программу, которая позволяет пользователю управлять банковским счетом (пополнение, снятие, просмотр баланса). Используйте функции для каждой операции.
5. **Игра «Сапер»**: Напишите программу, реализующую игру "Сапер". Используйте функции для генерации поля, отображения поля, обработки хода и проверки окончания игры.
6. **Программа для работы с матрицами**: Разработайте программу, которая выполняет основные операции с матрицами (сложение, вычитание, умножение). Используйте функции для каждой операции.
7. **Система управления библиотекой**: Создайте программу для управления библиотекой. Пользователь может добавлять, искать, редактировать и удалять книги. Используйте функции для каждой операции.
8. **Программа для решения квадратных уравнений**: Напишите программу, которая решает квадратные уравнения. Используйте функции для вычисления дискриминанта и корней уравнения.

Продвинутый уровень:

1. **Программа для работы с графами**: Разработайте программу, которая позволяет пользователю создавать, редактировать и анализировать графы. Используйте функции для добавления и удаления вершин и ребер, поиска кратчайшего пути и т.д.
2. **Программа для работы с большими числами**: Разработайте программу, которая выполняет арифметические операции с большими числами (больше, чем может хранить стандартный тип данных). Используйте функции для сложения, вычитания, умножения и деления.
3. **Программа для работы с регулярными выражениями**: Напишите программу, которая позволяет пользователю работать с регулярными выражениями (поиск, замена, проверка соответствия). Используйте функции для каждой операции.