

Практика 1

Задание 1

Необходимо отсортировать первые две трети списка в порядке возрастания, если среднее арифметическое всех элементов больше нуля; иначе — лишь первую треть.

Остальную часть списка не сортировать, а расположить в обратном порядке.

Задание 2

Написать программу «успеваемость». Пользователь вводит 10 оценок студента. Оценки от 1 до 12. Реализовать меню для пользователя:

- Вывод оценок (вывод содержимого списка);
- Пересдача экзамена (пользователь вводит номер элемента списка и новую оценку);
- Выходит ли стипендия (стипендия выходит, если средний балл не ниже 10.7);
- Вывод отсортированного списка оценок: по возрастанию или убыванию.

Задание 3

Написать программу, реализующую сортировку списка методом усовершенствованной сортировки пузырьковым методом.

Усовершенствование состоит в том, чтобы анализировать количество перестановок на каждом шагу, если это количество равно нулю, то продолжать сортировку нет смысла — список отсортирован.

Практика 2

Задание 1

Написать программу «справочник». Создать два списка целых. Один список хранит идентификационные коды, второй — телефонные номера. Реализовать меню для пользователя:

- Отсортировать по идентификационным кодам;
- Отсортировать по номерам телефона;
- Вывести список пользователей с кодами и телефонами;
- Выход.

Задание 2

Написать программу «книги». Создать два списка годы выпуска. Реализовать меню для пользователя:

- Отсортировать по названию книг;
- Отсортировать по годам выпуска;
- Вывести список книг с названиями и годами выпуска;
- Выход;

Практика 3

Задание 1

Есть четыре списка целых. Необходимо их объединить в пятом списке. Полученный результат в зависимости от выбора пользователя отсортировать по убыванию или возрастанию. Найти значение, введенное пользователем, с использованием линейного поиска.

Задание 2

Есть четыре списка целых. Необходимо объединить в пятом списке только те элементы, которые уникальны для каждого списка. Полученный результат в зависимости от выбора пользователя отсортировать по убыванию или возрастанию. Найти значение, введенное пользователем, с использованием бинарного поиска.

Практика 4

Задание 1

Написать программу, выполняющую сортировку списка целых чисел методом пузырьковой сортировки.

Задание 2

Написать программу, выполняющую сортировку списка целых чисел методом вставок.

Задание 3

Есть список целых. Необходимо первую половину списка отсортировать по убыванию, вторую половину по возрастанию.

Задание 4

Написать программу, выполняющую сортировку списка целых чисел методом слияния.

Практика 5

Задание 1

Написать программу, выполняющую сортировку списка целых чисел методом Шелла.

Задание 2

Написать программу, выполняющую сортировку списка целых чисел методом пирамидальной сортировки.

Задание 3

Написать программу, выполняющую сортировку списка целых чисел методом быстрой сортировки.

Задание 4

Есть стопка оладий различного радиуса. Единственная операция, проводимая с ними — между любыми двумя суем лопатку и меняем порядок оладий над лопаткой на обратный. Необходимо за минимальное количество операций таких отсортировать снизу вверх по убыванию радиуса.

Практика 6*

Задание 1

Есть список из 10 элементов, заполненный случайными числами. Необходимо найти число, введенное пользователем. Используйте алгоритм линейного поиска.

Задание 2

Есть список из 10 элементов, заполненный случайными числами. Необходимо найти число, введенное пользователем. Используйте алгоритм бинарного поиска.

*При решении задания можно дополнительно проверить тип передаваемого аргумента: принимается только строка.

Задание 3

Создайте класс Soda (для определения типа газированной воды), принимающий 1 аргумент при инициализации (отвечающий за добавку к выбираемому лимонаду).

В этом классе реализуйте метод `show_my_drink()`, выводящий на печать «Газировка и {ДОБАВКА}» в случае наличия добавки, а иначе отобразится следующая фраза: «Обычная газировка».

Задание 4

Николаю требуется проверить, возможно ли из представленных отрезков условной длины сформировать треугольник.

Для этого он решил создать класс `TriangleChecker`, принимающий только положительные числа.

С помощью метода `is_triangle()` возвращаются следующие значения (в зависимости от ситуации):

- Ура, можно построить треугольник!;
- С отрицательными числами ничего не выйдет!;
- Нужно вводить только числа!;
- Жаль, но из этого треугольник не сделать.

*Построить треугольник из отрезков можно лишь в одном случае: сумма длин двух любых сторон всегда больше третьей.

Задание 5

Николай – оригинальный человек.

Он решил создать класс Nikola, принимающий при инициализации 2 параметра: имя и возраст. Но на этом он не успокоился.

Не важно, какое имя передаст пользователь при создании экземпляра, оно всегда будет содержать “Николая”.

В частности - если пользователя на самом деле зовут Николаем, то с именем ничего не произойдет, а если его зовут, например, Максим, то оно преобразуется в “Я не Максим, а Николай”.

Более того, никаких других атрибутов и методов у экземпляра не может быть добавлено, даже если кто-то и вздумает так поступить (т.е. если некий пользователь решит прибавить к экземпляру свойство «отчество» или метод «приветствие», то ничего у такого хитреца не получится).

Задание 6

Рассмотрим объект «Программист», который задаётся именем, должностью и количеством отработанных часов. Каждая должность имеет собственный оклад (заработную плату за час работы). В нашей импровизированной компании существуют 3 должности:

Junior — с окладом 10 тугриков в час;

Middle — с окладом 15 тугриков в час;

Senior — с окладом 20 тугриков в час по умолчанию и +1 тугрик за каждое новое повышение.

Напишите класс Programmer, который инициализируется именем и должностью (отработка у нового работника равна нулю). Класс реализует следующие методы:

work(time) — отмечает новую отработку в количестве часов time;

rise() — повышает программиста;

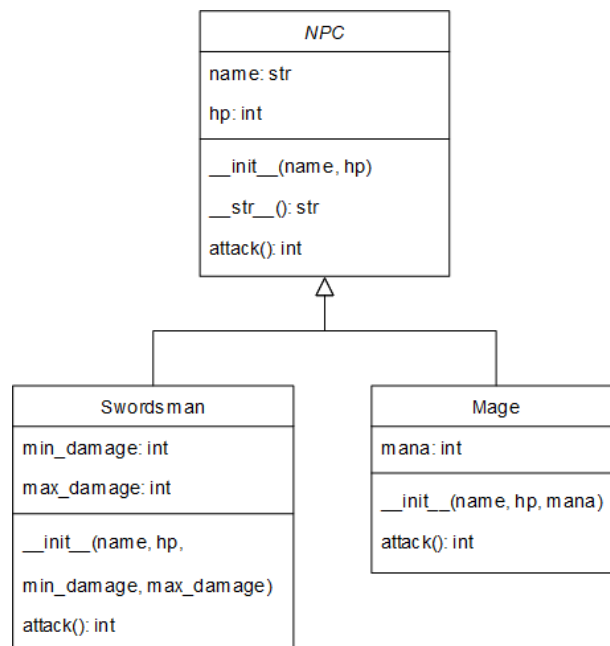
info() — возвращает строку для бухгалтерии в формате: <имя> <количество отработанных часов>ч. <накопленная зарплата> тгр.

Задание 7

Опишите ряд классов, необходимых для реализации противников в компьютерной игре:

- `NPC` – базовый класс
- `Swordsman` – мечник (наследует класс `NPC`). Наносит случайный урон из заданного диапазона.
- `Mage` – маг (наследует класс `NPC`). Наносит урон равный удвоенному количеству маны. Не может атаковать, если мана закончилась.

Диаграмма классов приведена на следующем рисунке:



Пример:

Explain

Имя: Бильбо, Очки здоровья: 15

Не могу атаковать!

Имя: Гендальф, Очки здоровья: 100

Маг Гендальф нанёс 10 урона!

Имя: Арагорн, Очки здоровья: 50

Мечник Арагорн нанёс 8 урона!

Имя: Гендальф, Очки здоровья: 100

Не могу атаковать! Мана закончилась.

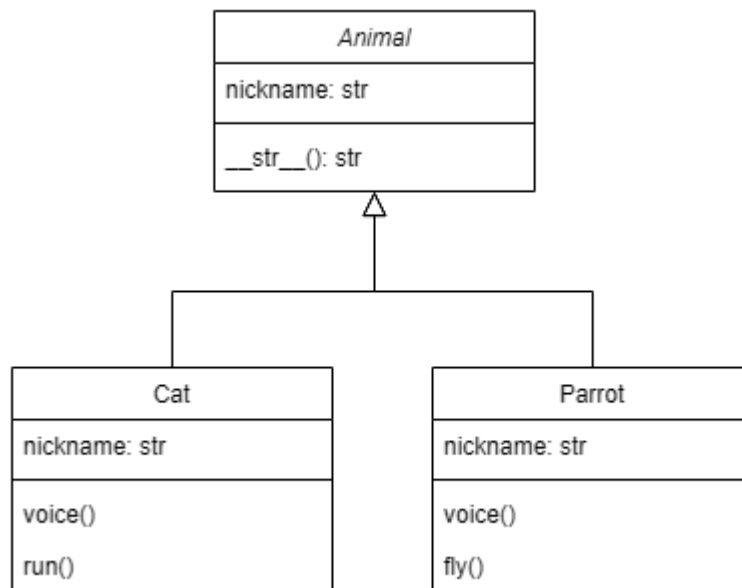
Практика 7*

Задание 1

Дан класс `Animal`, который содержит поле `nickname` (кличка), конструктор и метод `__str__`.

Опишите классы `Cat` (Кот) и `Parrot` (Попугай), которые будут наследниками класса `Animal`. Оба класса должны содержать методы `voice()`, который выводит на экран фразу произнесённую животным. Класс `Cat` должен содержать метод `run()` (бежать), который выводит на экран фразу *Побежали!*. Класс `Parrot` должен содержать метод `fly()` (летать), который выводит на экран фразу *Полетели!*.

Диаграмма классов изображена на следующем рисунке:



Задание 2

Опишите класс `Message`, который описывает абстрактное сообщение, которое можно отправить в мессенджере. Класс содержит следующие поля:

- `sender` - отправитель сообщения (строка)
- `recipient` - получатель сообщения (строка)

Добавьте к описанию класса следующие методы:

- `__init__(sender, recipient)` - конструктор класса

Задание 3

Опишите класс `MSDice` который реализует многогранную игральную кость. Описание класса содержит количество граней и текущее значение.

Дополните класс для кубиков D4, D6, D10, D20.

Задание 4

Создайте класс `Player`, описывающий игрока у которого есть следующие поля:

- `nickname` - имя игрока (строка)
- `exp_points` - количество очков опыта (целое число); начальное значение - 0
- `inventory` - список предметов, которые есть у игрока (список); начальное значение - []

Класс должен содержать следующие методы:

- `__init__(nickname)` - конструктор, принимающий в качестве параметра имя игрока. Также конструктор инициализирует поля `exp_points=0` и `inventory=[]`
- `__str__` - преобразование объекта в строку (формат вывода см. в примере)
- `addExp(exp)` - добавить игроку `exp` очков опыта
- `addItem(item)` - добавить к списку `inventory` предмет `item` (строка)
- `removeItem(item)` - удалить из списка `inventory` предмет с именем `item` (строка)

Задание 5

Создайте класс `resistors`, который содержит следующие функции:

- `parallel(r1, r2)` - вычисляет общее сопротивление двух резисторов, соединённых параллельно
- `consec([r1, r2, r3, ...])` - вычисляет общее сопротивление резисторов, соединённых последовательно. В качестве параметра функция принимает список чисел.

При параллельном соединении двух резисторов с сопротивлением r_1 и r_2

общее сопротивление вычисляется по формуле:

$$R_{\text{пар}} = \frac{r_1 \cdot r_2}{r_1 + r_2}$$

При последовательном соединении резисторов, общее сопротивление равно сумме отдельных сопротивлений:

$$R_{\text{пол}} = r_1 + r_2 + r_3 + \dots$$