Лабораторная работа 4

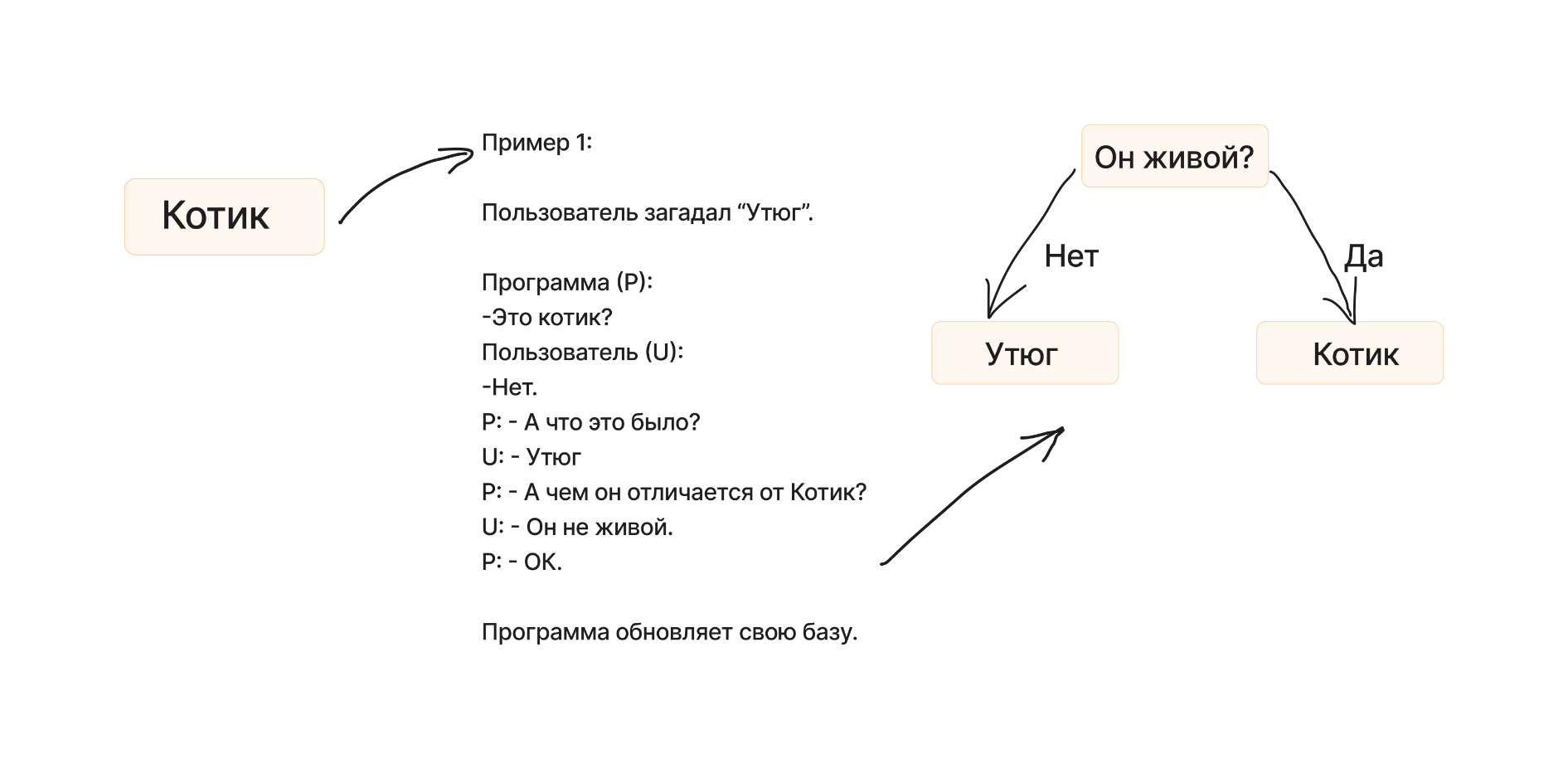
**Цель**: освоить принципы работы с бинарными деревьями.

**Задача**: написать программу-аналог Акинатора.

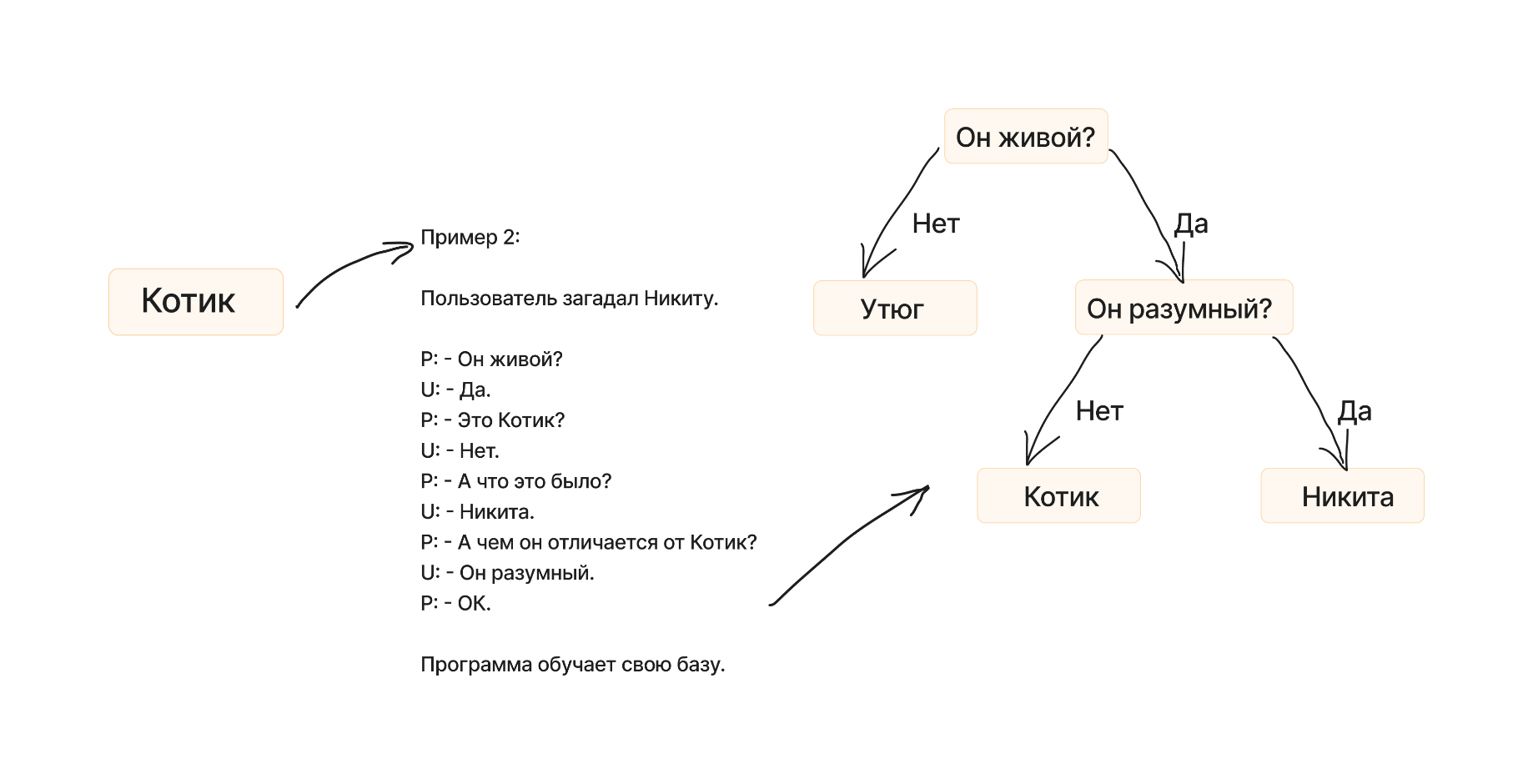
**Условия:**

Программа должна определять загаданный пользователем объект. Для этого она должна управлять двоичным деревом, в котором хранится информация о загадываемых объектах. Работа проходит по следующему сценарию.

1. Пользователь загадывает объект.
2. Программа задает вопросы, на которые пользователь отвечает да/нет.
3. Каждый ответ приводит к очередному вопросу из дерева. Так до тех пор, пока не будет достигнут конец (лист) с ответом.
4. Если ответ неверный, программа должна предложить ввести данные о загаданном объекте (отличительный вопрос) и запомнить новый объект.



Примечание: Программа становится осмысленнее, если работает с предметами какой-то ограниченной тематики, иначе вопросы в дереве будут странными и хаотичными. Тематику выбирайте сами.



**На 4-6**

- Описать отдельное API (библиотечку) для работы с двоичным деревом.

- Размер дерева не должен ограничиваться хардкодом (дефайнами, константами, максимальным размером, введенным с клавиатуры и т.п.), то есть дерево должно иметь возможность расти безгранично.

**На 7-8**

- Реализовать функцию сохранения / загрузки базы данных об объектах (сохранение дерева в файл). Чтобы не вносить базу каждый раз заново.

- Подготовить тестовую базу на объектов 20.

**На 9-10**

- Допилить какую-нибудь интересную фичу на свой вкус)

[как вариант, можно сделать логирование]

**Условия приема**:

* Оформление **-** (названия файлов, функций, переменных, отступы, структура проекта и т.п.) и небольшие баги должны быть исправлены. // Code name convention (camelCase, snake\_case)
* Использование Git (использование файлов gitignore, README)
* Условие автомата: Sonar (Отсутствие багов и code Smells)