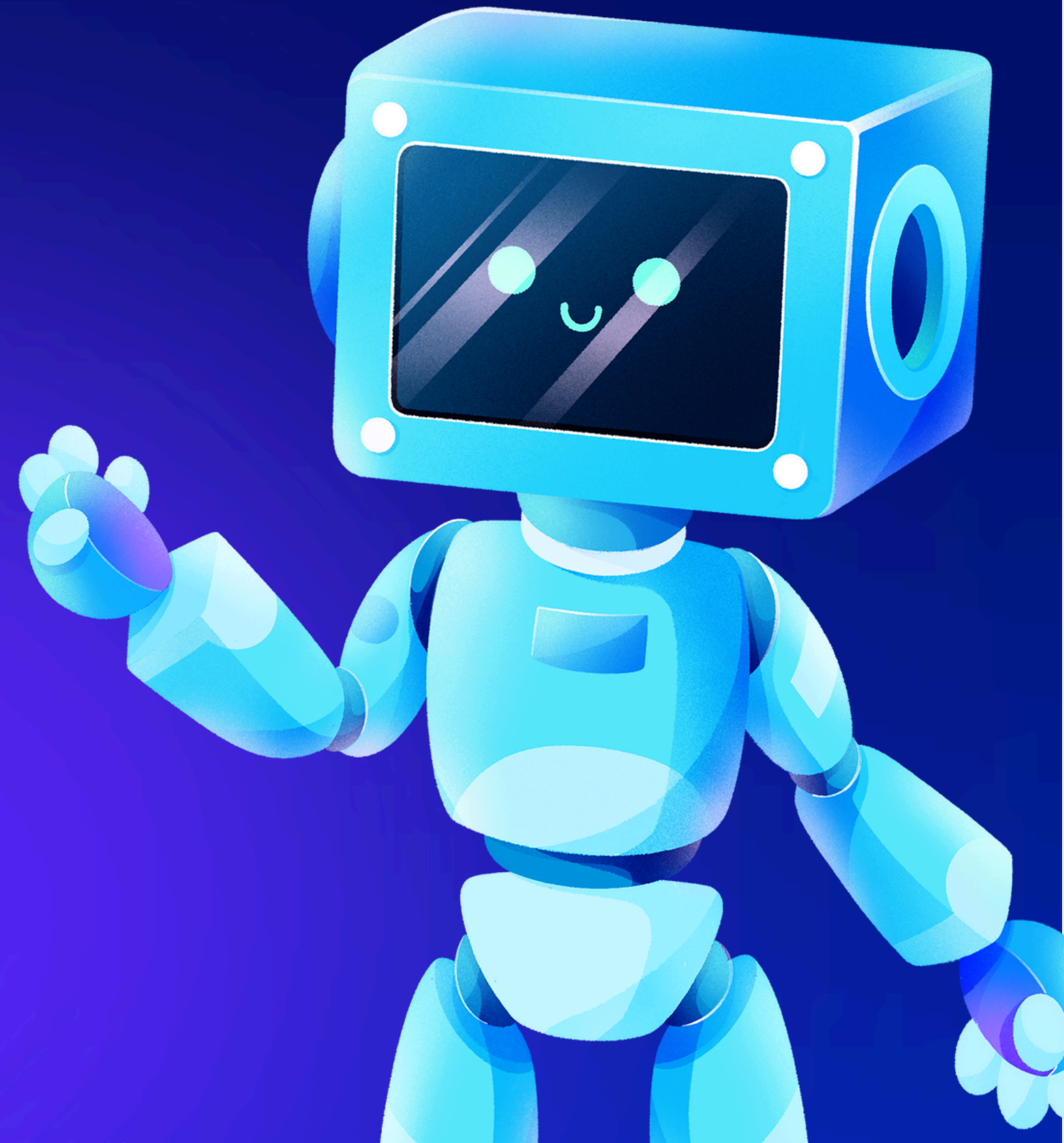
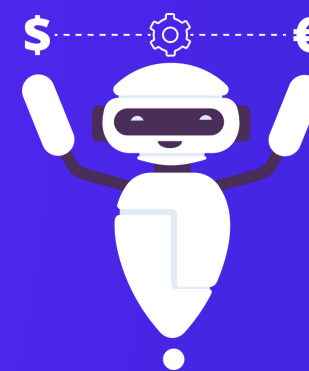
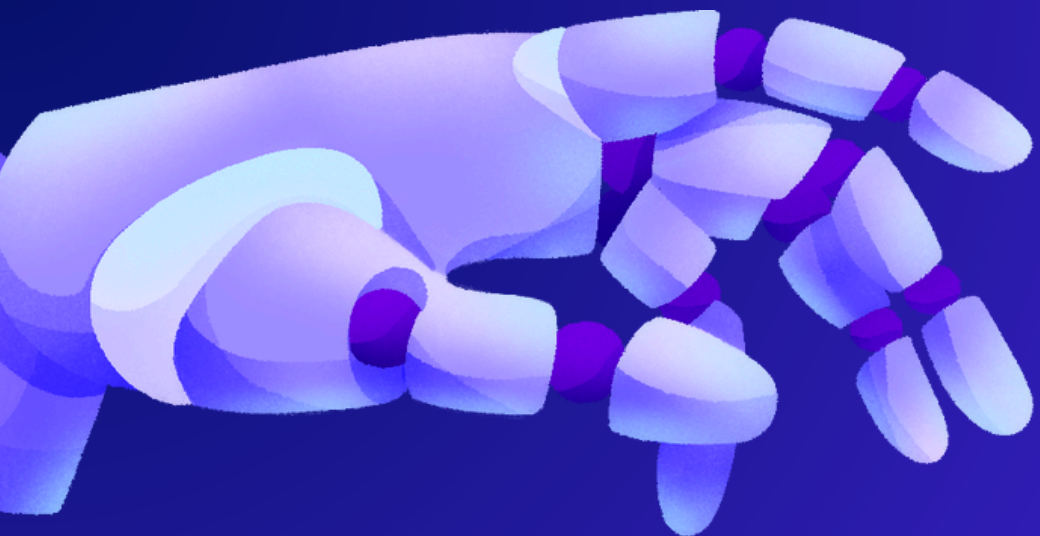


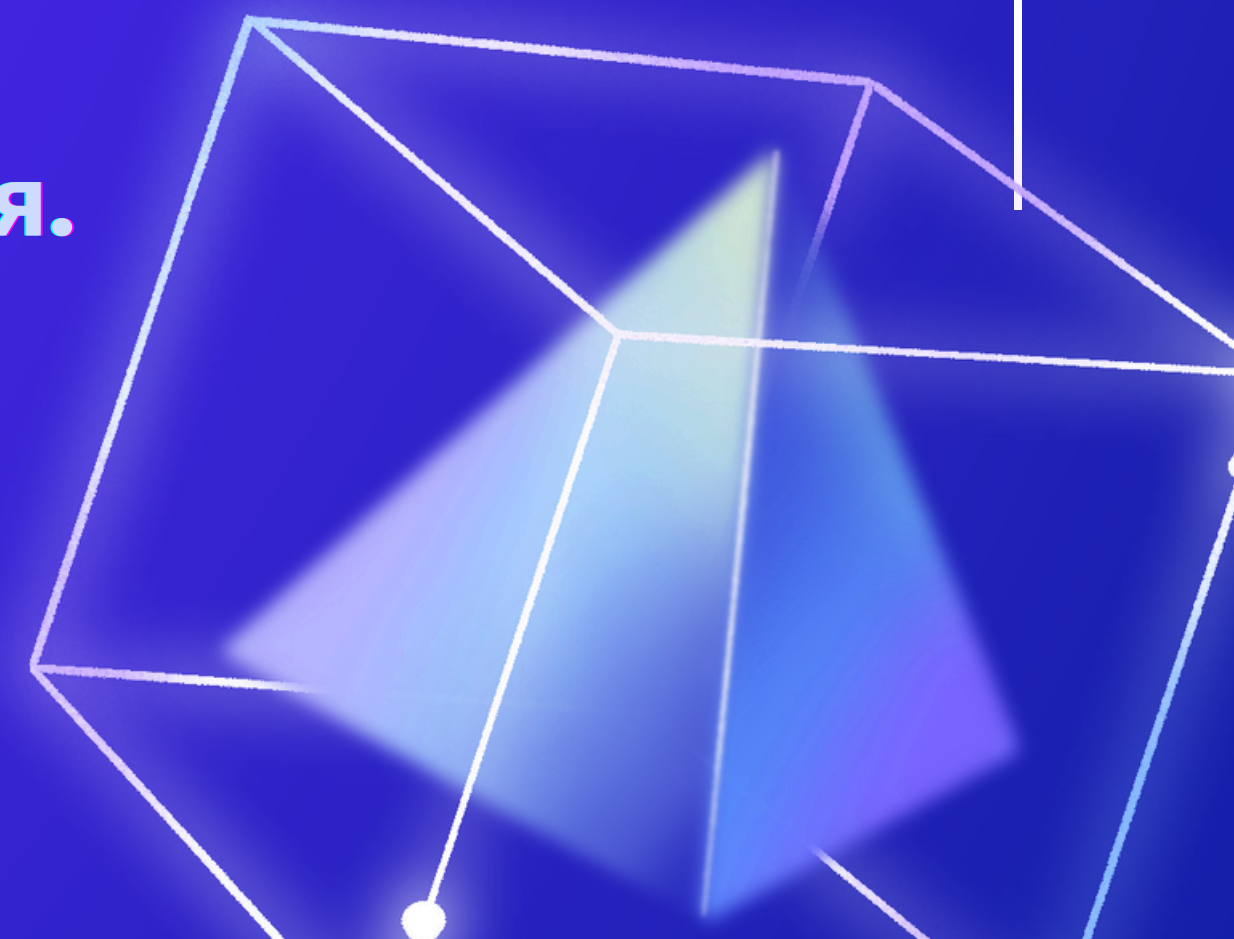
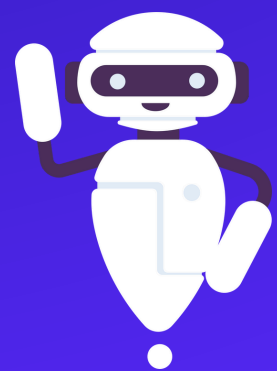
Критические ситуации в рекурсии

выполнили: "Коды круглые"





Критические ситуации в рекурсии возникают, когда рекурсивная функция не имеет условия выхода или когда условие выхода не удовлетворяется, что приводит к бесконечному рекурсивному вызову. Это может привести к переполнению стека или другим ошибкам времени выполнения.



Рассмотрим пример:



```
#include <iostream>
using namespace std;

double Depos(double sum, double p, int n) {
    if (n == 0) {
        return sum;
    } else {
        double proc = sum * p / 100.0;
        sum += proc;
        return Depos(sum, p, n - 1);
    }
}

int main() {
    double sum, p;
    int n;

    cout << "Введите сумму вклада: ";
    cin >> sum;

    cout << "Введите процент: ";
    cin >> p;

    cout << "Введите количество периодов времени: ";
    cin >> n;

    double fDepos = Depos(sum, p, n);

    return 0;
}
```

РАЗБОР

В данной программе единственная потенциальная критическая ситуация, которая может возникнуть, это бесконечный рекурсивный вызов.



Рекурсивная функция `Depos` уменьшает значение `n` на единицу при каждом рекурсивном вызове.

Однако, если пользователь введет отрицательное значение `n`, функция будет вызывать сама себя бесконечно, поскольку условие выхода `n == 0` никогда не будет достигнуто.

Чтобы избежать этой критической ситуации, следует добавить проверку на отрицательные значения `n` и выводить соответствующее сообщение об ошибке, если `n` отрицательно.

СПАСИБО

