

Постановка задачи:

Внесите изменения в модель так, чтобы появился хотя бы еще один уровень, связанный с другими элементами модели. Далее протестируйте обновленную модель, внесите изменения, если будет необходимо. Полученную модель преобразуйте в код алгоритмического языка по рассмотренной 22 ноября методике. Полученный код измените так, чтобы программа предлагала вводить данные перед каждым шагом моделирования, предварительно выводя основные параметры объекта моделирования.

Описание модели:

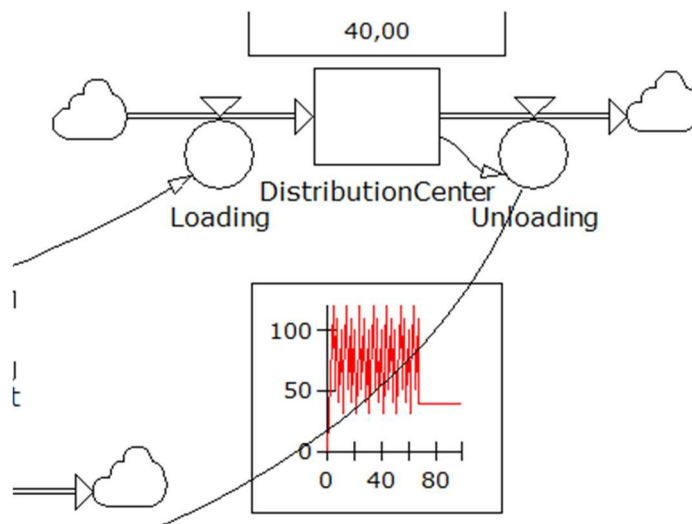
В данной модели я добавила следующий функционал

- учет перевозки товара через уровень (в машину погрузили, затем выгрузили - промежуточный между складами уровень);

Ход работы:

Модель была изменена следующим образом:

- 1) Добавлен новый уровень учёт перевозки товара на распределительный склад



Здесь переменная TransferActualVolume теперь идет не в GoodsTransfer, а во входную переменную нового уровня Loading.

Далее, в уровне DistributionCenter товар и как только товара накопится ≥ 100 , грузовик уезжает (переменная Unloading) в

ShopStore. Тем самым, перевозка осуществляется только по 100 единиц товара до тех пор, пока товар на складе не закончится.

Также добавлен график наличия товара на промежуточном складе.

Измененный код на python

```
#include <iostream>
#include <stdio.h>

double Level_1, Level_2, Level_3, Level_Intermediate, Rate_1, Rate_2, Rate_3,
Rate_4, Rate_Intermediate, Constant_1, Constant_2, Constant_3, Auxiliary_1,
Auxiliary_2, dt;

void inputValues() {
    std::cout << "Enter Level_1, Level_2, Level_3, Constant_1, Constant_2,
Constant_3, dt: ";
    std::cin >> Level_1 >> Level_2 >> Level_3 >> Constant_1 >> Constant_2 >>
Constant_3 >> dt;
}

int main()
{
    inputValues(); // Ввод начальных значений

    //Инициализация переменных и темпов
    Auxiliary_2 = 14;
    Auxiliary_1 = Constant_1 + Constant_2 + Constant_3 * Level_1;
    Rate_1 = Level_1 - Level_2;
    Rate_2 = Level_1 + Auxiliary_1;
    Rate_3 = Level_2 * 0.1;
    Rate_4 = Auxiliary_2 * 0.2;

    printf("Init:\n");
    printf("%10.2f %10.2f %10.2f %10.2f %10.2f %10.2f %10.2f\n", Level_1, Level_2,
Level_3, Rate_1, Rate_2, Rate_3, Rate_4);

    //Пошаговый расчет
    for (int i = 0; i < 5; i++)
    {
        inputValues(); // Ввод значений перед каждым шагом

        // Передвижение товаров между уровнями и складами
        double move_to_intermediate = Level_2 >= 100 ? 100 : Level_2; // Перемещение
в промежуточный склад, не более 100 единиц
        Level_Intermediate += move_to_intermediate;
        Level_2 -= move_to_intermediate;

        double move_to_store = Level_Intermediate >= 100 ? 100 : Level_Intermediate;
// Перемещение на магазин со склада, не более 100 единиц
        Level_Intermediate -= move_to_store;
        Level_3 += move_to_store;

        // Обновление уровней
        Level_1 = Level_1 - dt * Rate_2 + dt * Rate_1;
        Level_2 = Level_2 - dt * Rate_3 + dt * Rate_2;
        Level_3 = Level_3 - dt * Rate_4 + dt * Rate_3;

        // Обновление переменных и темпов
        Auxiliary_1 = Constant_1 + Constant_2 + Constant_3 * Level_1;
        Rate_1 = Level_1 - Level_2;
```

```
Rate_2 = Level_1 + Auxiliary_1;
Rate_3 = Level_2 * 0.1;
Rate_4 = Auxiliary_2 * 0.2;

printf("Step %d: ", i + 1);
printf("%10.2f %10.2f %10.2f %10.2f %10.2f %10.2f %10.2f\n", Level_1,
Level_2, Level_3, Rate_1, Rate_2, Rate_3, Rate_4);
    }
}
```