**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.

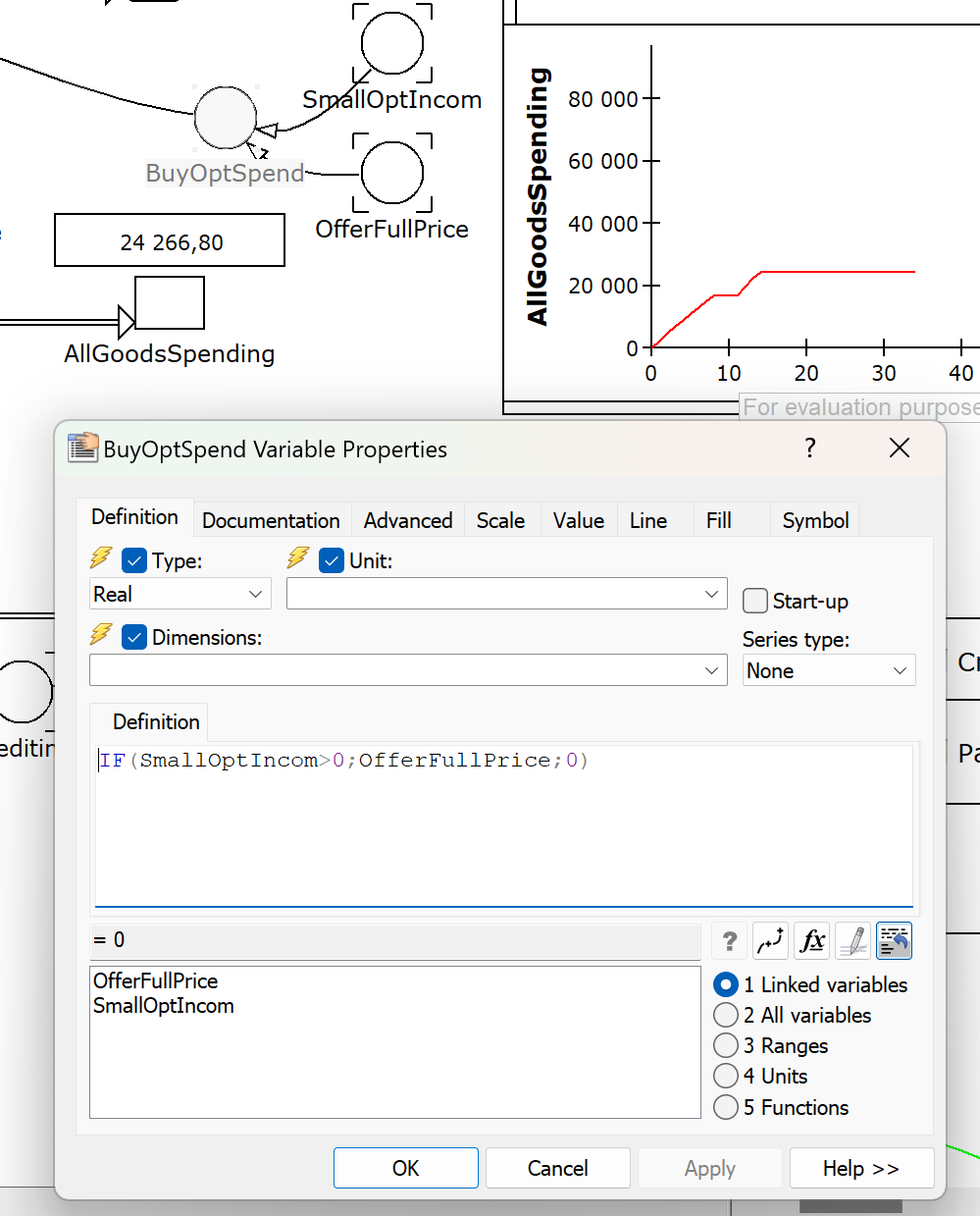
**Описание модели:**В данной модели я добавила следующий функционал

* Возможность кредитования магазина;

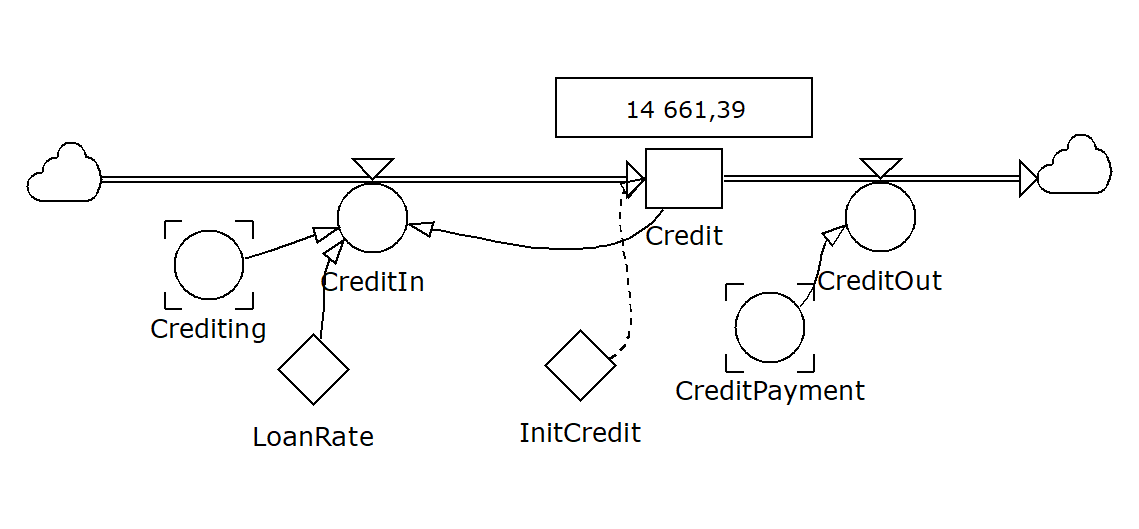
**Ход работы:**

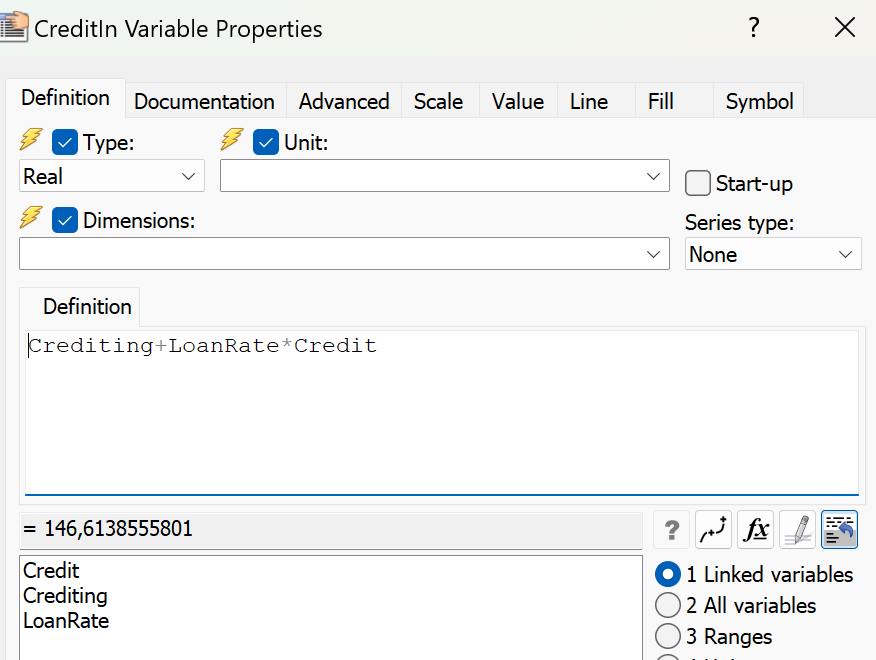
Модель была изменена следующим образом:

1. Восстановлены уровень и темпы для оплаты закупаемых товаров



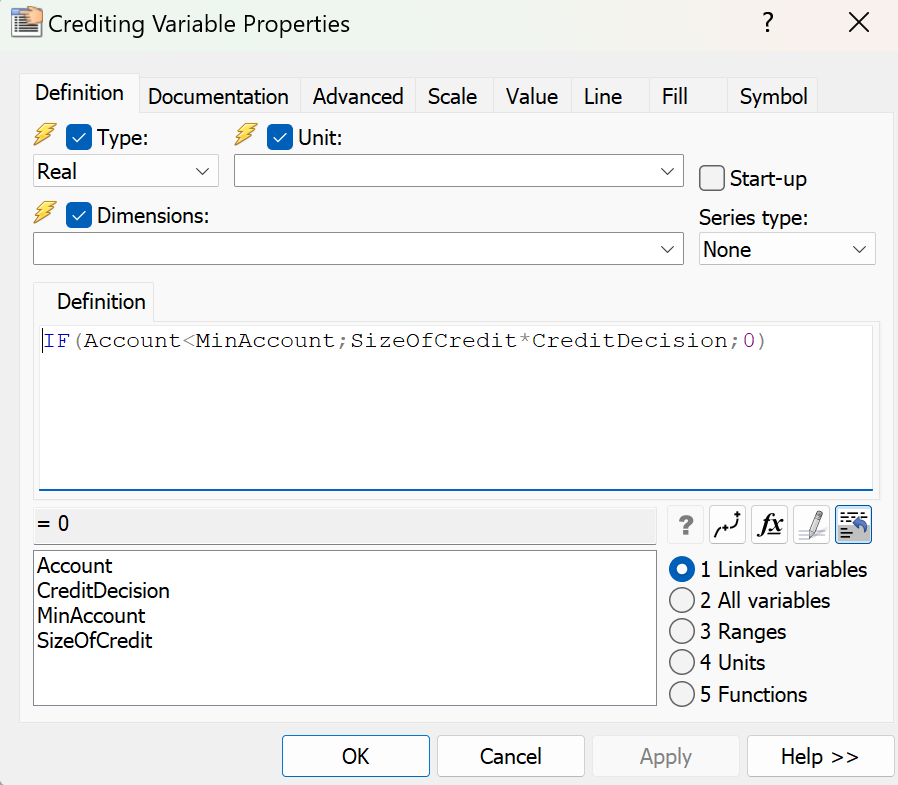
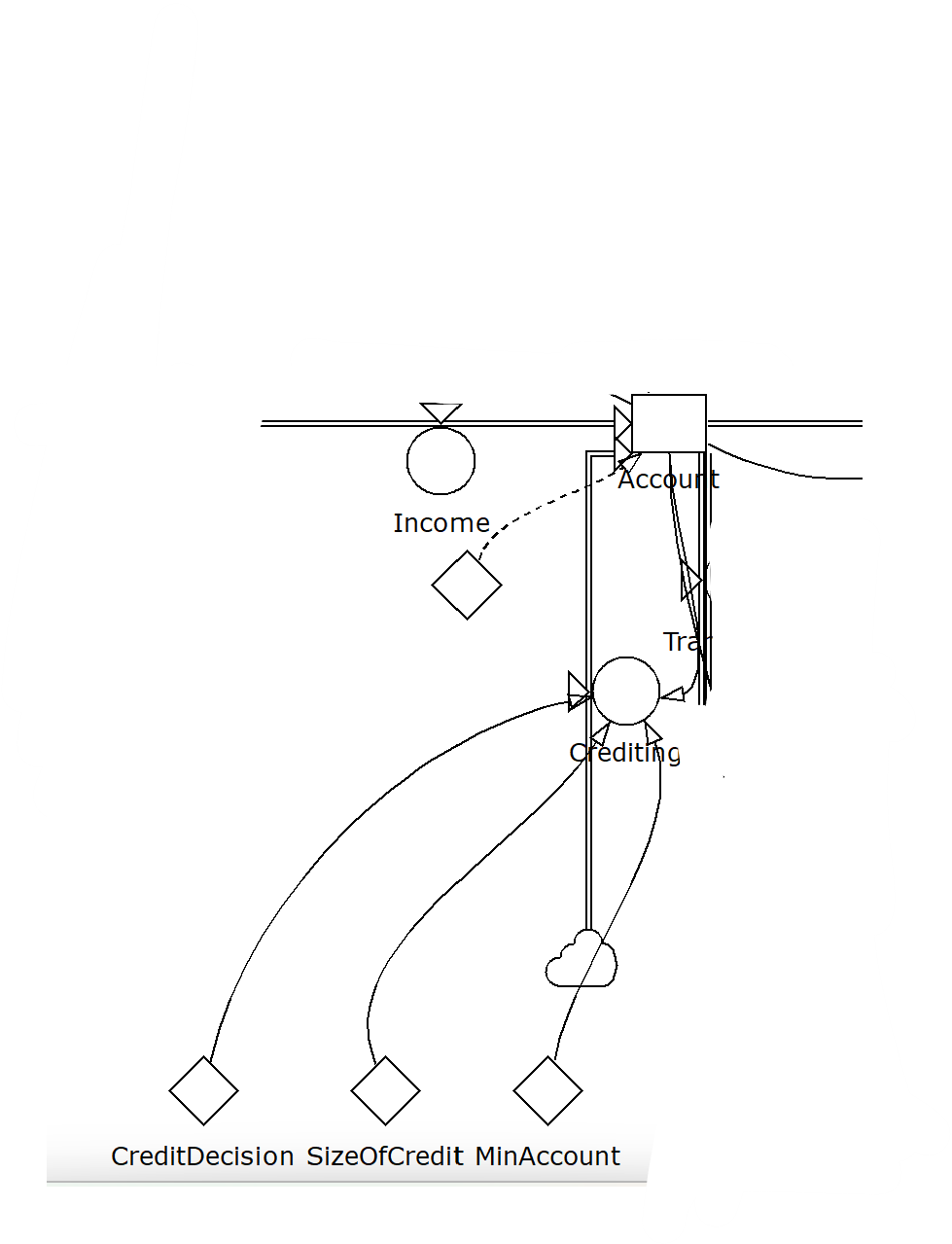
1. Добавлен новый уровень учёта текущего кредита Credit. Он увеличивается при взятии кредита магазином (переменная Crediting) и при учитывании процентной ставки (константа LoanRate). Уровень уменьшается при погашении кредита магазином (переменная CreditPayment)





1. Переменная Crediting увеличивает счет Account и зависит от констант:

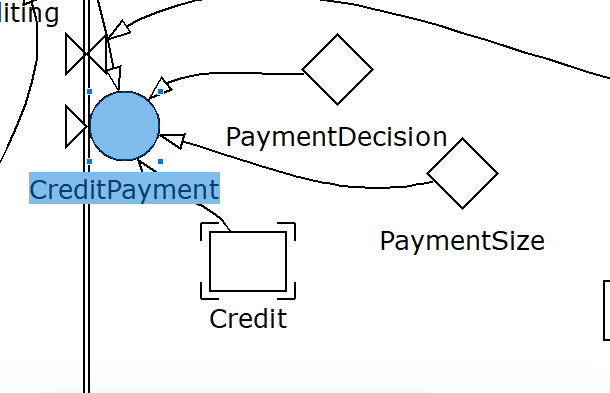
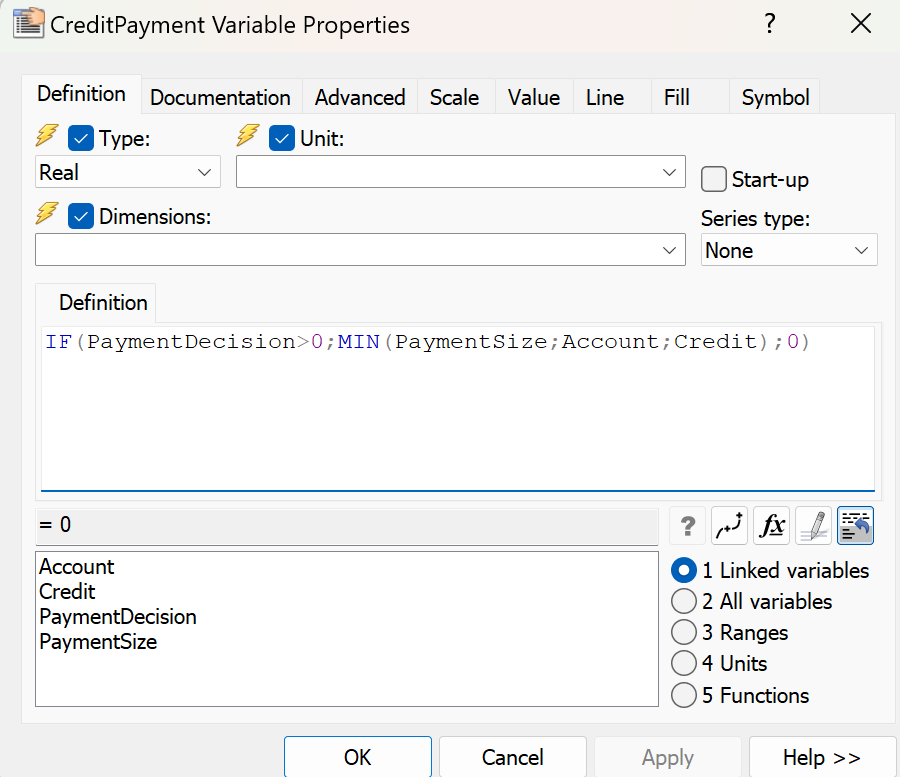
* CreditDecision – решение о взятии кредита (0 – нет, 1 -да);
* SizeOfCredit – размер взятого кредита;
* MinAccount – счет меньше которого можно брать кредит;

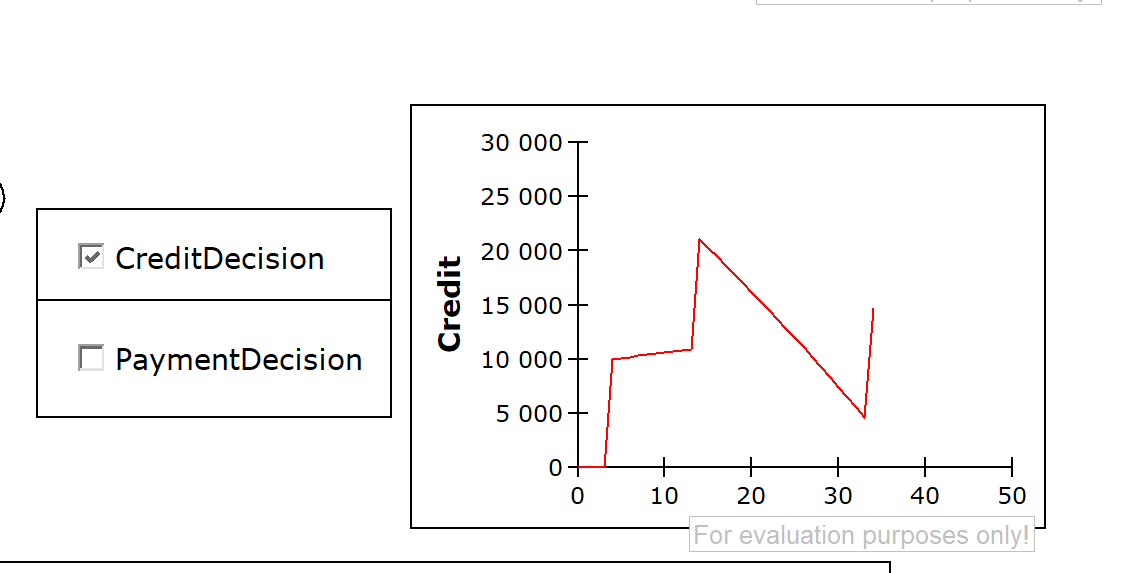


1. Переменная CreditPayment уменьшает счет Account и зависит от констант:

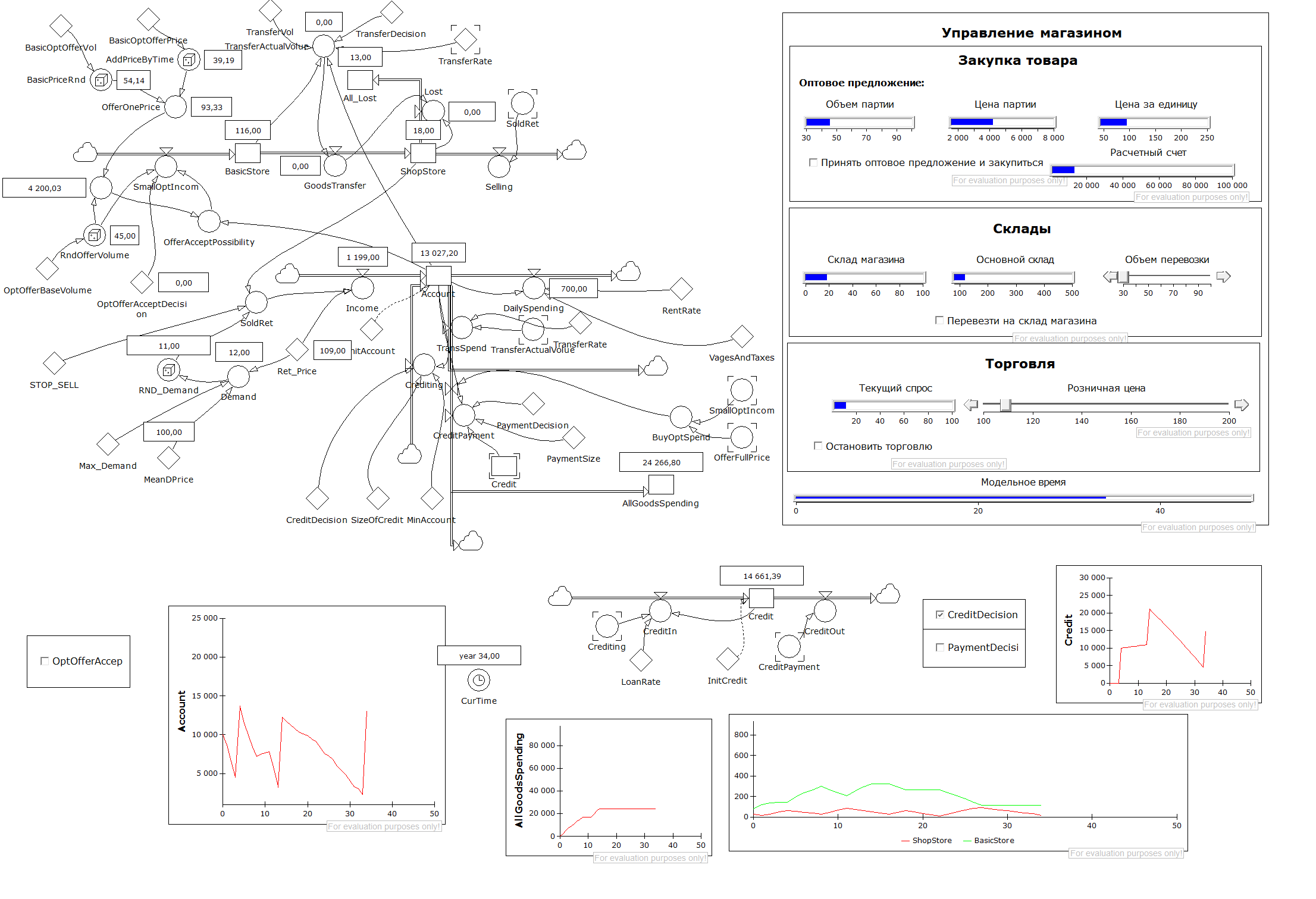
* PaymentDecision – решение о погашении кредита (0 – нет, 1 -да)
* PaymentSize – размер выплаты

Также переменная CreditPayment не может быть больше текущего долга Credit и счета Account.

1. Добавлен график текущего кредита и элементы Checkbox для принятия решений о взятии/погашения кредита. 

Модель полностью:



**Измененный код на c++**

#include <iostream>

#include <stdio.h>

using namespace std;

//Объявление уровней и констант

double credit, crediting, creditPayment, loanRate, account, sizeOfCredit, minAccount, paymentSize;

int creditDecision, paymentDecision;

int main(){

setlocale(LC\_ALL, "Russian");

//Инициализация уровней и констант

creditDecision = 0; paymentDecision = 0;

credit = 0; loanRate = 0.01;

account = 10000; sizeOfCredit = 10000;

minAccount = 5000; paymentSize = 1000;

creditPayment = 1000;

//Пошаговый расчет

for (int t = 0; t < 10; t++)

{

creditDecision = 0; paymentDecision = 0;

//вывод данных

printf("Шаг: %d Счет: %10.2f Текущий кредит: %.2f\n",t, account, credit);

//ввод данных

if (account < minAccount) {

cout << "Желаете взять кредит?(1 - да, 0 - нет) ";

cin >> creditDecision;

if (creditDecision) {

cout << "Размер кредита: ";

cin >> sizeOfCredit;

}

}

if (credit > 0) {

cout << "Желаете выплатить часть кредита? (1 - да, 0 - нет) ";

cin >> paymentDecision;

if (paymentDecision) {

cout << "Размер выплаты: ";

cin >> paymentSize;

}

}

//изменение уровней, переменных и темпов

account = account + (rand() % 3000) - 2000; //доход-убыль магазина

crediting = sizeOfCredit \* creditDecision;

creditPayment = min(account, min(paymentSize \* paymentDecision, credit));

account = account + crediting - creditPayment; //кредитование

credit = credit + (credit \* loanRate) + crediting - creditPayment;

}

system("pause");

}

