



Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра алгоритмических языков

Отчёт по заданию практикума

# **Моделирование работы животноводческой фермы**

**Работу выполнил:**

студент 425 группы

Пономаренко Вячеслав Сергеевич

Москва, 2024

# Оглавление

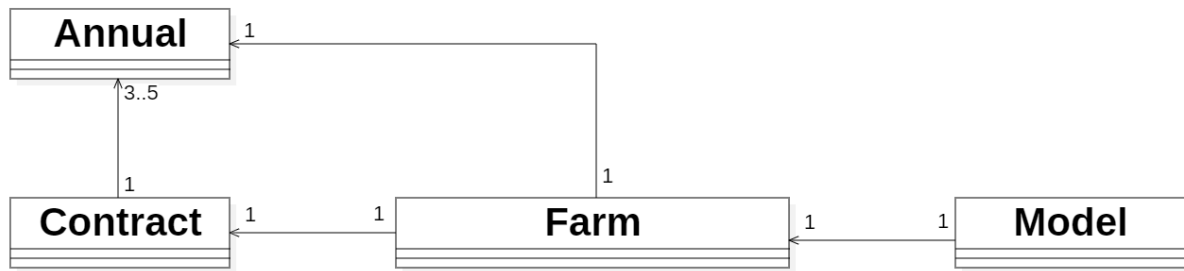
Уточнение постановки задачи .....	3
Диаграмма классов.....	4
Спецификации интерфейса.....	4
Диаграмма объектов .....	7
Инструментальные средства.....	7
Описание файловой структуры программы .....	7
Пользовательский интерфейс.....	8

## Уточнение постановки задачи

Задача - создать модель, симулирующую работу животноводческой фермы:

- Симуляция происходит в периоде от 3 до 5 лет;
- В начале каждого года должно быть закуплено корма ровно на сумму, указанную в контракте на этот год;
- В случае неспособности купить корма на указанную сумму симуляция заканчивается банкротством фермы из-за невыполнения условий контракта;
- В конце каждого года должно быть продано ровно столько молодых, взрослых и старых животных, сколько указано в контракте на этот год;
- В случае недостатка тех или иных животных, за каждого недостающего платится фиксированная неустойка, указанная в контракте на этот год;
- В случае неспособности выплатить неустойку, симуляция заканчивается банкротством фермы из-за невыполнения условий контракта;
- Все условия контракта на каждый год указываются пользователем;
- Кол-во животных и капитал фермы на начало симуляции указываются пользователем;
- Цена корма, необходимого одному взрослому животному на год, указывается пользователем;
- Вероятность несчастных случаев и возможная смертность скота от них указывается пользователем.

## Диаграмма классов



## Спецификации интерфейса

*//интерфейс класса, представляющего эксперимент*

class Model:

Public:

*//конструктор класса Model*

\_\_init\_\_(self)

*//возвращает оставшееся время контракта (в годах)*

get\_rem\_years(self)

*//возвращает номер текущего года*

get\_current\_year(self)

*//возвращает полную длительность заключенного контракта*

get\_duration(self)

*//возвращает число молодых животных на ферме*

get\_num\_young(self)

*//возвращает число взрослых животных на ферме*

get\_num\_adult(self)

*//возвращает число старых животных на ферме*

get\_num\_old(self)

*//возвращает текущий капитал фермы*

get\_capital(self)

*//сбор информации из полей графического интерфейса*

```

collect_information(self, contract_duration, ... , yearly_food)
//осуществление шага симуляции (симуляция одного года)
step(self)
//интерфейс класса, представляющего ферму
class Farm:
Public:
    //конструктор класса Farm
    __init__(self, yo, ad, ol, ... , alpha=1.5, beta=1, gamma=0.9, ro=0.3)
    //возвращает номер текущего года
    get_current_year(self)
    //возвращает число молодых животных на ферме
    get_num_young(self)
    //возвращает число взрослых животных на ферме
    get_num_adult(self)
    //возвращает число старых животных на ферме
    get_num_old(self)
    //возвращает текущий капитал фермы
    get_capital(self)
    //закупка корма на указанную в контракте сумму
    buy_food(self)
    //изменение численности скота при недостатке корма
    death_starvation(self)
    //изменение численности скота после размножения
    reproduction(self)
    //продажа животных по контракту (и выплата неустойки)
    sell_animals(self)
    //симуляция одного года

```

```

        simulate_one_year(self)

//интерфейс класса, представляющего полный контракт
class Contract:
    Public:
//конструктор класса Contract
        __init__(self, contract_duration, ... , cost_food_5, penalty_5)
//возвращает условия контракта на указанный год
        get_year_contract(self, year)
//возвращает полную длительность заключенного контракта
        get_duration(self)

//интерфейс класса, представляющего условия контракта на один
определённый год
class Annual:
    Public:
//конструктор класса Annual
        __init__(self, yo, yo_price, ad, ad_price, ol, ol_price, f_cost, pen)
//кол-во молодых на продажу в этом году
        get_num_young(self)
//кол-во взрослых на продажу в этом году
        get_num_adult(self)
//кол-во старых на продажу в этом году
        get_num_old(self)
//цена молодых в этом году
        get_price_young(self)
//цена взрослых в этом году
        get_price_adult(self)
//цена старых в этом году
        get_price_old(self)

```

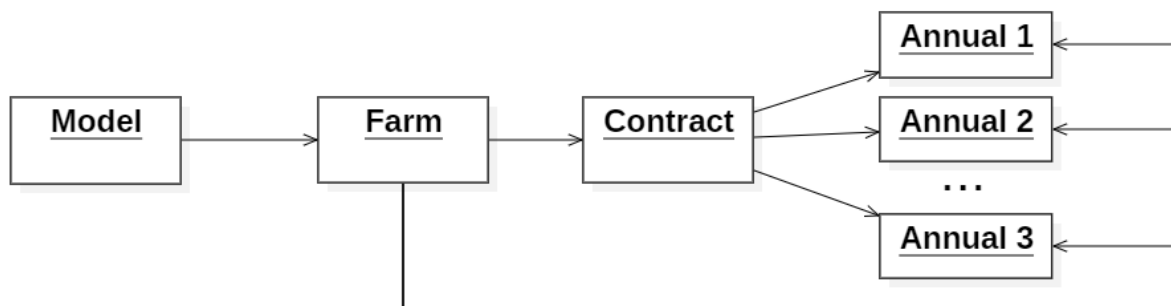
*//сумма, на которую нужно закупить корм в этом году*

`get_price_food(self)`

*//неустойка за единицу скота в этом году*

`get_penalty(self)`

## Диаграмма объектов



## Инструментальные средства

- Язык разработки: Python;
- Среда разработки: Jupiter Notebook;
- Библиотеки: tkinter.

## Описание файловой структуры программы

Farm.py - файл, содержащий в себе все классы (Model, Farm, Contract, Annual) и построение графического интерфейса.

# Пользовательский интерфейс

**Начальные данные**

Молодые: 50 ед.  
Взрослые: 50 ед.  
Старые: 50 ед.  
Капитал: 2000 \$

Годовая цена корма на взрослого: 10 \$

Вероятность несчастных случаев: 15 %  
Умрёт от 5 % до 20 %

Длительность контракта: 5

1-ый год 2-ой год 3-ий год 4-ый год 5-ый год

**Животные на продажу**

Молодые:		Взрослые:		Старые:	
Кол-во	Цена за ед.	Кол-во	Цена за ед.	Кол-во	Цена за ед.
10 ед.	50 \$	10 ед.	100 \$	10 ед.	10 \$

Цена корма: 1000 \$  
Неустойка: 50 \$

СТАРТ  
ШАГ  
ДО КОНЦА  
ЗАНОВО  
ВЫХОД

Пользовательский интерфейс включает в себя возможность определить начальные условия фермы (кол-во молодых, взрослых и старых животных на ферме, начальный капитал), на какую сумму потребляет корма одно взрослое животное в год, вероятность происхождения несчастных случаев и какая часть животных может от них погибнуть.

Также он позволяет определить длительность контракта фермы (3-5 лет) и условия контракта на каждый отдельный год.

При нажатии на кнопку “СТАРТ”:

- Очищается текстовое поле вывода информации по симуляции;
- Происходит сбор данных из окон ввода (пустые поля расцениваются как нулевые значения);
- Блокируются окна ввода;
- Разблокируются кнопки “ШАГ”, “ДО КОНЦА”, “ЗАНОВО”;
- Блокируется кнопка “СТАРТ”.



При нажатии на кнопку “ШАГ”:

- Происходит симуляция одного года работы фермы;
- В текстовое поле выводится информация по тому, как прошёл год;
- Если текущий год был последним по контракту или была достигнута ситуация банкротства, симуляция завершается, кнопки “ШАГ” и “ДО КОНЦА” блокируются.

При нажатии на кнопку “ДО КОНЦА”:

- Воспроизводятся шаги симуляции фермы до условий завершения симуляции;
- Кнопки “ШАГ” и “ДО КОНЦА” блокируются.

При нажатии на кнопку “ЗАНОВО”:

- Симуляция прекращается;
- Кнопки “ШАГ”, “ДО КОНЦА” и “ЗАНОВО” блокируются.
- Разблокируется кнопка “СТАРТ”

При нажатии на кнопку “ВЫХОД”:

- Работа программы прекращается.