

# Описание разработанных классов

## Distribution

Класс предназначен для получения числовых данных, необходимых для проведения моделирования. Все сложные математические расчёты проводятся в этом классе

### Описание атрибутов

mu	int	Атрибуты, задающие параметры логистического распределения	
S	int		
clients	int	Задаёт желаемое число покупателей	
RNG	Random	Класс для получения случайных значений	
IdealSum	double	Хранит точную сумму нескольких значений функции распределения	Нужны для более точного соответствия заданного и полученного числа покупателей
RealSum	int	Хранит сумму нескольких округлённых значений функции распределения	
TrickyScale	float	Задаёт оптимальный масштаб для распределения количества товара	

### Описание операций

double Logistic(double Median, double Scale, double x)	Функция, вычисляющая плотность вероятности логистического распределения в точке.	Median	Параметры распределения
		Scale	
		x	Точка, для которой вычисляется функция
float Tricky(float count, float Scale, float min)	Функция, вычисляющая количество товара	count	Среднее количество товара
		Scale	Масштаб
		min	Минимальное количество товара
int GetIntCount(int count)	Функция, возвращающая количество упакованного товара	count	Среднее количество упаковок

float GetFloatCount(float count, float min)	Функция, возвращающая количество развесного товара	count	Среднее количество товара
		min	Минимальное количество товара
int GetBuyers(int time)	Функция, возвращающая количество покупателей, пришедших в заданный промежуток времени	time	Номер промежутка времени

# SQLAgent

Класс предназначен для работы с базой данных MySQL

## Описание атрибутов

DBName	String	Имя базы данных
URL	String	Сетевой адрес базы данных
user	String	Имя пользователя
password	String	Пароль
conn	Connection	Подключение к базе данных

## Описание операций

void HandleEx(SQLException ex)	Функция обработки исключений SQL	ex	Исключение
void Connect()	Функция, создающая подключение к БД		
void Disconnect()	Функция, закрывающая подключение к БД		
boolean TestConnect()	Функция, проверяющая правильность параметров подключения к БД		
int Started(int peoplecount, int goodscount)	Функция, создающая новую запись о симуляции. Возвращает id симуляции	peoplecount	Заданное пользователем количество людей
		goodscount	Заданное пользователем среднее количество товаров
boolean LoadSettings()	Загружает настройки подключения к БД. Возвращает true в случае успеха		

void Ended(int peoplearrived, int peopleserved, float avggoodscount, float avgprofit, int profit, int maxqueue, int maxqueueuetime, boolean iscorrect, int simid)	Функция, обновляющая запись о симуляции	peoplearrived	Количество пришедших покупателей
		peopleserved	Количество обслуженных покупателей
		avggoodscount	Среднее количество купленных товаров
		avgprofit	Средняя выручка
		profit	Выручка
		maxqueue	Максимальная длина очереди
		maxqueueuetime	Омер временного интервала, когда длина очереди была максимальной
		incorrect	Правильно ли посчитана выручка
		simid	id симуляции
void Bought(int buyerid, int time, int simid, Buyer buyer)	Функция, записывающая информацию о покупках одного покупателя	buyerid	Номер покупателя
		time	Время покупки
		simid	id симуляции
		buyer	Покупатель, информация о покупках которого заносится в БД
LinkedList<Product> GetProductInfo()	Функция, возвращающая полную информацию об ассортименте магазина		
LinkedList<String> ShowProductInfo()	Функция, возвращающая полную информацию об ассортименте магазина в удобном для пользователя виде		

LinkedList<String > GetResults(int simulationid)	Функция, возвращающая подробную информацию о симуляции	simulationid	Номер симуляции
boolean DeleteProduct(int code)	Функция, удаляющая запись о товаре из базы данных	code	Код удаляемого товара
boolean AddProduct(Produ ct item)	Функция, добавляющая информацию о новом продукте	item	Продукт, информацию о котором надобно добавить
UpdateProduct(Pr oduct item)	Функция, обновляющая информацию о продукте	item	Продукт, описывающий изменения
void Fix()	Функция, исправляющая сбойные записи о симуляциях		

## Terminal

Класс, представляющий собой модель кассового терминала с кассиром.  
Обслуживает покупателей

### Описание атрибутов

id	int	Порядковый номер обслуживаемого покупателя
Money	int	Общее количество полученных денег
Profit	int	Выручка от продаж
simid	int	id проводимой симуляции
AmICorrect	boolean	Переменная, показывающая, были ли проведённые денежные расчёты корректными

### Описание операций

void Serve(Buyer buyer, int time)	Функция, обслуживающая покупателя и заносщая информацию о нём в отчёт	buyer	Покупатель, который обслуживается
		time	Временной промежуток, в течение которого покупатель обслуживается

## Admin

Класс, являющийся моделью менеджера торгового зала. Выдаёт базовую сумму после сдачи выручки и проверяет корректность в конце рабочего дня

### Описание атрибутов

Money	int	Количество имеющихся денег
Amount	int	Общая сумма выданных денег

### Описание операций

boolean CheckGig(int gig)	Функция проверки корректности работы терминала	gig	Выручка терминала
int getMoney()	Функция-getter для поля Money	См. определения геттеров и сеттеров	
setMoney(int Money)	Функция-сеттер для поля Money		
getAmount()	Функция-getter для поля Amount		
setAmount(int Amount)	Функция-сеттер для поля Amount		

## Buyer

Класс, представляющий модель покупателя

### Описание атрибутов

ArriveTime	int	Номер временного интервала, в течение которого покупатель встал в очередь
ServedTime	int	Номер временного интервала, в течение которого обслуживание покупателя было завершено
Money	int	Количество имеющихся с собой денег
Discount	boolean	Показывает, есть ли у покупателя дисконтная карта
Cart	LinkedList<Product>	Корзина с продуктами

### Описание операций

void AddProduct(Product thing)	Функция, добавляющая в корзину покупателя товар и некоторую сумму (не меньше необходимой) в кошелёк	thing	Добавляемый товар
int GetTotal()	Функция, возвращающая суммарную стоимость всех товаров в корзине с учётом наличия скидки		



## Generator

Класс, создающий новых покупателей и набор товаров для каждого из них

### Описание атрибутов

RangeOfGoods	LinkedList<Product>	Ассортимент магазина
AVGGoodsCount	int	Среднее количество товаров (по наименованиям), приобретаемое покупателями
size	int	Количество товаров, продающихся в магазине

### Описание операций

Generator(int avggoodscount)	Конструктор	avggoodscount	Среднее количество товаров (по наименованиям), приобретаемое покупателями
Buyer CreateBuyer()	Функция, создающая нового покупателя		

## Stat

Класс, реализующий сбор статистики

### Описание атрибутов

PeopleServed	int	Количество обслуженных людей
PeopleArrived	int	Количество пришедших людей
MaxQueue	int	Максимальная длина очереди
MaxQueueTime	int	Временной интервал, в течение которого длина очереди впервые достигла своего максимума
AvgGoodsCount	float	Среднее количество купленных товаров
AvgProfit	float	Средняя выручка
Profit	int	Общая выручка
Goods	int	Атрибут, являющийся суммой количеств разновидностей товаров покупателей симуляции

### Описание операций

void Consider(Buyer cl)	Функция учёта информации о покупателе	cl	Покупатель, подлежащий учёту
-------------------------	---------------------------------------	----	------------------------------

## Product

Класс, представляющий из себя модель продукта. Не портится!

### Описание атрибутов

Name	String	Название товара	
Code	int	Код товара	
Count	float	Количество товара	
IsPacked	boolean	Показывает, упакован ли товар	Упакованные товары продаются только целыми
Price	float	Стоимость одной единицы товара	

### Описание операций

int GetTotalPrice()	Функция, возвращающая цену продукта		
---------------------	-------------------------------------	--	--

# TTerminal

Класс, непосредственно взаимодействующий с исследователем

## Описание атрибутов

workers	LinkedList<Worker>	Список проводимых симуляций
IsConfigured	boolean	Атрибут, показывающий были ли настроены параметры подключения к БД

## Описание операций

void MakeHeader(Print Writer out, String title, boolean error)	Функция, печатающая заголовок страницы	out	Класс вывода
		error	Надобно ли напечатать сообщение об ошибке?
void MakeFooter(Print Writer out)	Функция, печатающая окончание страницы	out	Класс вывода
void NoConnection(Pri ntWriter out)	Функция, печатающая сообщение об ошибке в случае невозможности установить соединение с базой данных	out	Класс вывода
void init()	Срабатывает при развёртывании	Функции унаследованы, аргументы являются стандартными	
void doGet(HttpServlet Request request, HttpServletResponse response)	Обрабатывает GET-запросы		
void doPost(HttpServlet Request request, HttpServletResponse response)	Обрабатывает POST-запросы		

## Worker

Класс, выполняющий симуляцию

### Описание атрибутов

steps	int	Атрибут, показывающий, какой из временных интервалов обрабатывается
ready	boolean	Атрибут, показывающий завершена ли симуляция
distr	Distribution	Для выполнения сложных расчётов
BuyersList	LinkedList<Buyer>	Очередь покупателей
generator	Generator	Генератор покупателей
stat	Stat	Статистика
simid	int	id симуляции
timer	Timer	Таймер
terminal	Terminal	Терминал+кассир
admin	Admin	Менеджер торгового зала
al	ActionListener	Обработчик событий таймера

### Описание операций

void StartSim(int peoplecount, int goodscount)	Функция, начинающая симуляцию	peoplecount	Примерное количество людей	Задаются исследовател ем
		goodscount	Среднее количество товаров	