



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
М. В. ЛОМОНОСОВА

Факультет вычислительной математики и кибернетики
Кафедра алгоритмических языков

Отчёт о выполнении задания практикума

«Моделирование обслуживания в филиале банка»

Студента 424 группы
Н. О. РОЖКОВ

Москва, 2020

1 Постановка задачи

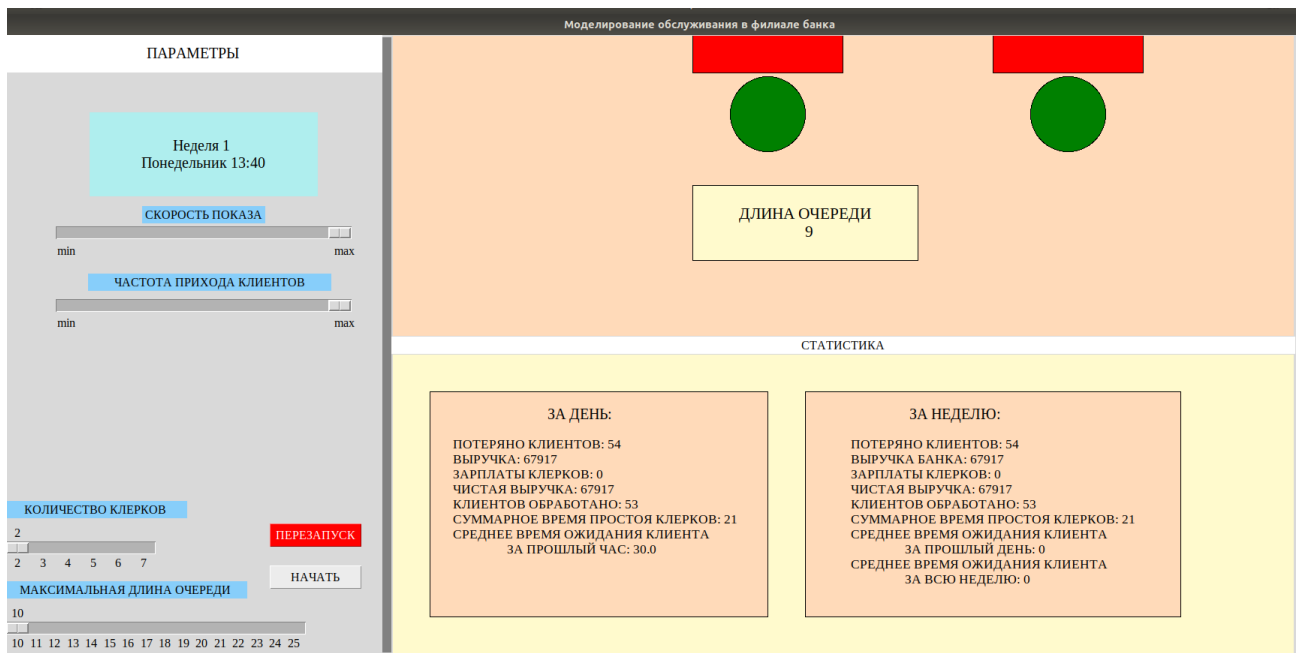
Необходимо создать компьютерную модель обслуживания потока заявок, поступающих от клиентов банка, несколькими клерками ($2 \leq N \leq 7$) в одном из филиалов банка. Известно недельное расписание работы филиала банка: 5 дней по 8 часов и один день – 7 часов. Реализовать возможность изменения параметров, таких как частота прихода клиентов, скорость течения времени. Сбор статистики за неделю и за каждый день.

1.1 Базовые требования

1. Интуитивно понятный пользовательский интерфейс.
2. Проектирование в рамках ООП.
3. Визуализация работы клерков.
4. Интерфейс для изменения параметров моделирования.
5. Визуализация результатов моделирования.
6. Использование актуальных библиотек.
7. Написание кода, используя PEP8.

1.2 Пользовательский интерфейс

1. Изменение скорости показа работы модели.
2. Изменение скорости подачи заявок.
3. Настройку количества клерков (от 3 до 7).
4. Настройку максимальной длины общей очереди (от 10 до 25).



2 Классы проекта

Проект состоит из следующих классов:

- **ForConst** — константы;
- **ExternalWorld** — класс создает модель внешнего мира, т.е. создает банк генерирует заявки клиентов и отвечает за все что изменяется со временем;
- **Interface** — Отрисовка состояния моделирования;
- **BankBranch** — Модель филиала банка, распределяет клиентов по клеркам;

В классе **ForConst** описаны следующие константные значения:

- `WIDTH` и `HEIGHT` — размер игрового поля;
- `WEEK` — Дни недели;
- `LEN_MIN` и `LEN_HOUR` и `LEN_WEEK` — Количество минут, часов и дней;
- `SALARY` — Зарплата клерка за один день;
- `SHORT_WORK_DAY` — Сокращенный рабочий день;
- `NOT_WORK_DAY` — Не рабочий день;
- `END_WORK_DAY` — Время конца рабочего дня;
- `START_WORK_DAY` — Время начала рабочего дня;
- `END_SHORT_DAY` — Время окончания короткого рабочего дня;
- `COEF_NORM_TIME_SPEED` — Коэффициент нормального течения времени;
- `STEP_COEF_NORM_TIME_SPEED` — Шаг изменения времени;
- `LEN_SHORT_WORK_DAY` — Длина короткого рабочего дня;
- `LEN_WORK_DAY` — Длина рабочего дня;
- `HOURLY_WHEN_LINES_CLIENTS_WILL_MORE` — Час начиная с которого поток клиентов увеличивается;
- `NOT_WORK_DAY_NUM` — Номер не рабочего дня;

В классе `ExternalWorld` описаны следующие методы:

- `__init__` — создает внешний мир;
- `format_clock_text` — нормализация времени;
- `tick` — отвечает за изменение состояния моделирования с течением времени;
- `what_is_coef_speed_time` — определяет коэффициент скорости прихода клиентов в зависимости от времени и дня недели;
- `make_klerks` — считывает количество клерков и отображает их;
- `delete_clients` — удаляет клиентов;
- `restart` — перезапуск моделирования;

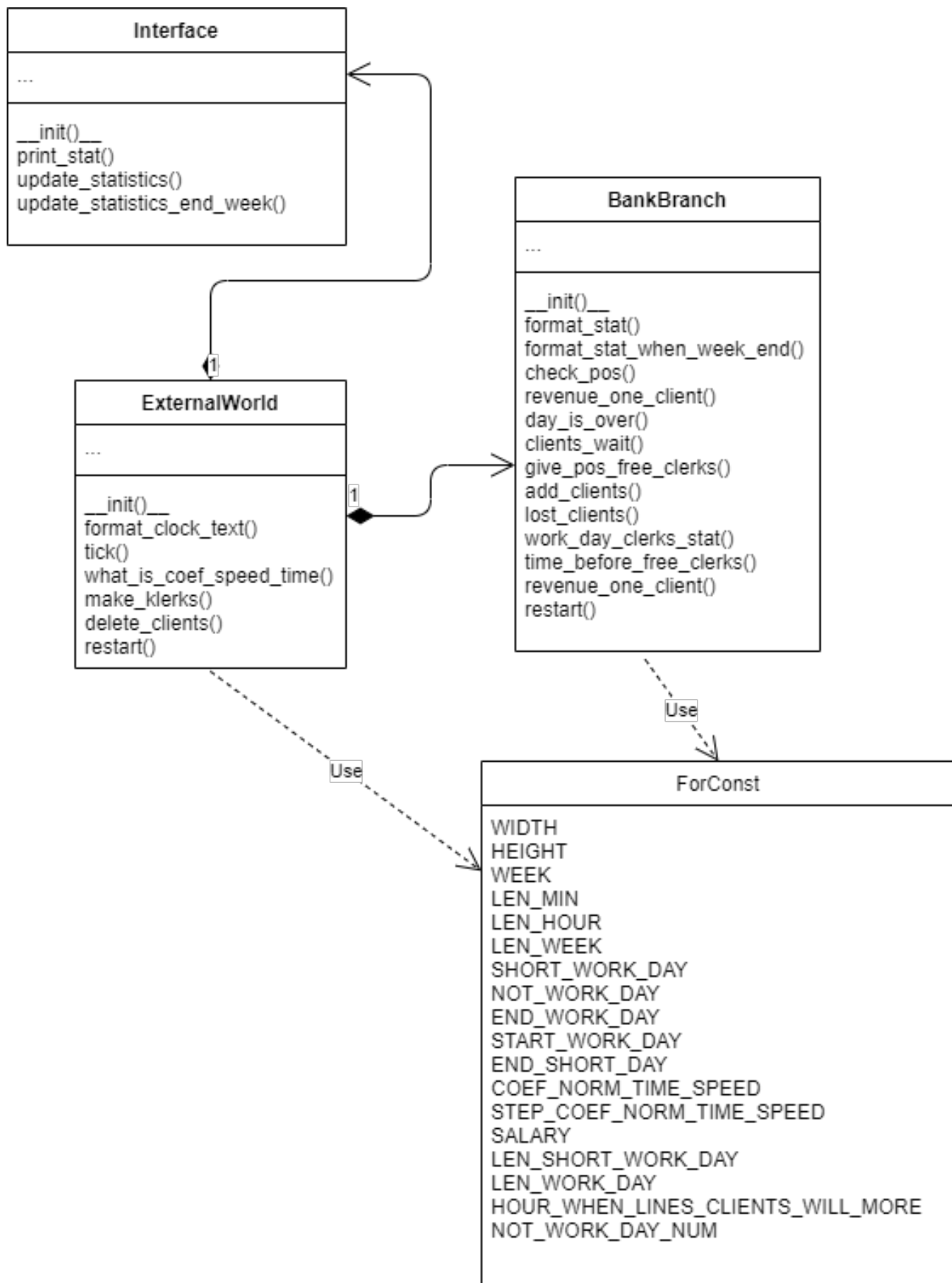
В классе `BankBranch` реализованы следующие методы:

- `__init__` — создание банка;
- `formatsstat` — — — ; `formatsstatwhenweekend` — — — ; `dayisover` — — — ; `client`
- `check_pos` — определение свободного клерка;
- `revenue_one_client` — определяет выручку полученную от одного клиента;

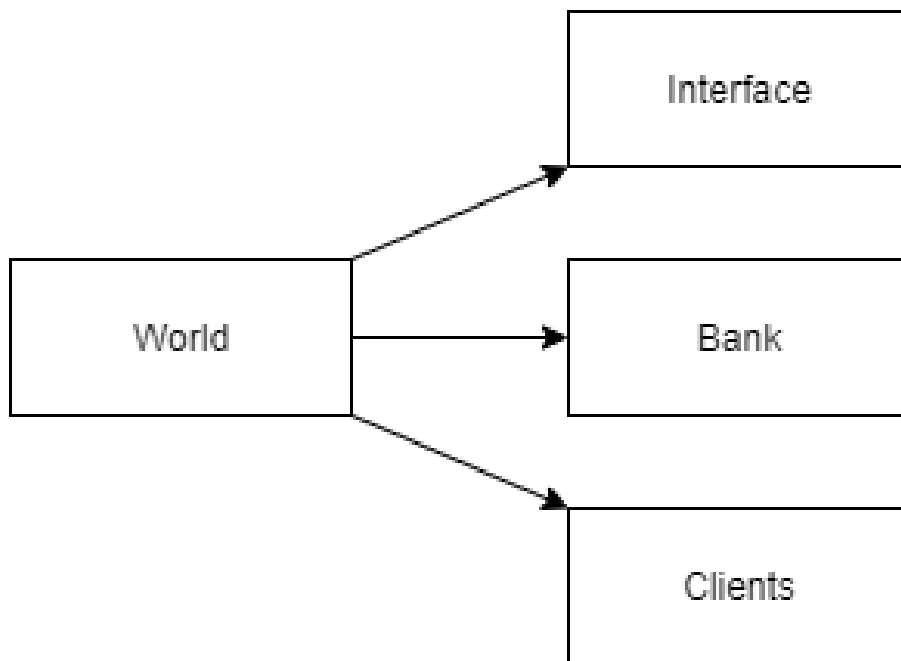
В классе `Interface` реализованы следующие методы:

- `__init__` — визуализация всей модели;
- `print_stat` — отрисовка статистики;
- `update_statistics` — обновление статистики;
- `update_statistics_end_week` — обновление статистики в конце недели;

3 Диаграмма классов



4 Диаграмма объектов



5 Используемые библиотеки

При реализации использовались следующие библиотеки:

- `tkinter` — графический интерфейс и обработка внешних событий.
- `random` — генерация случайных чисел.
- `time` — работа со временем.

6 Сценарии работы с приложением

Приложение запускается при помощи команды `python3 <Название>.py`. Необходимо выбрать интересующую вас конфигурацию модели используя ползунки и нажать кнопку "НАЧАТЬ" для начала моделирования. Далее на экране появятся красные прямоугольники - это клерки, в часы работы возле них появляются красные шарики - это клиенты. Под ними расположен прямоугольник в котором отображается длина очереди в данный момент.

В нижней части располагаются прямоугольники в которых отображается статистика за день и за неделю.

Для изменения конфигурации моделирования после запуска нажмите кнопку "ПЕРЕЗАПУСК". Затем измените интересующие вас параметры и нажмите кнопку "НАЧАТЬ".

7 Тестирование

Тестирование проводилось при различном количестве клерков 2, 5, 7 и скорости подачи заявок. В результате тестирования было выявлено следующее:

- Число потерянных клиентов при высокой скорости подачи заявок значительно отличается для разного количества клерков, если взять 2 клерка и максимальную длину очереди то мы теряем около 1200 клиентов и обрабатываем 400 за неделю.
- Выручка банка всегда положительная при данном моделировании так как зарплаты клерков несопоставимы с прибылью банка.
- Количество обработанных клиентов по сути и является основополагающим фактором, при максимальной длине очереди 10, что наиболее приближено к действительности как мне кажется при количестве клерков 7 число обработанных клиентов около 1200 а потерянных около 350.
- Время простоя очень сильно зависит от плотности потока клиентов, если плотность высокая то даже при 7 клерках их суммарный простой за неделю 500 минут, это конечно не сравнимо с 15 минутами у 2 клерков, но это совсем не реалистично звучит.
- Время ожидания клиента как и все остальное варьируется от количества клерков, длины очереди и плотности заявок. В пределах от 0 до 65 минут.

В итоге, по полученным результатам можно судить что если в банке высокая проходимость то лучше поставить больше клерков, а если не слишком высокая то можно обойтись силами и 2-3 клерков.