Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**О творческой работе**

Выполнил работу

студент группы РИС-20-2б

Савельев В.А.

Проверила

Доцент кафедры ИТАС

к.т.н Полякова О. А.

Пермь, 2020

**Цель работы**

Создание графических приложений:

* Калькулятор кредитов
* Визуальное решение задачи Коммивояжера

**Калькулятор кредитов**

**Дизайн:**

Дизайн пытается передать собой визуальную составляющую денежной купюры используя при этом цвета и форму приложения.

Все используемые цвета были выбраны исходя из триады цветового треугольника основанного на соотношении чисел Фибоначчи т.к. считается что любые пропорции и соотношения основанные на числах Фибоначчи наиболее приятные для человеческого глаза.

Триада цветового треугольника была использована и для комбинаций цветов основного (зеленого) спектра и для смежных цветов (светло-красный и светло-оранжевый).

Стоит обратить внимание, что все соотношения сторон и размеров приложения вместе с его элементами так же взяты из соотношения Фибоначчи. В купе чего даже весьма контрастное сочетание цветов выглядит гармонично и приятно с визуальной точи зрения.

Иконка и название приложения так же объявлены.

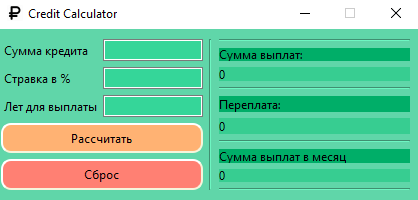


Рисунок 1: Интерфейс приложения

**Функционал:**

Интерфейс представляет собой 3 (три) поля:

* «Сумма кредита»
* «Ставка в процентах»
* «Количество лет для выплаты кредита»

А также кнопок:

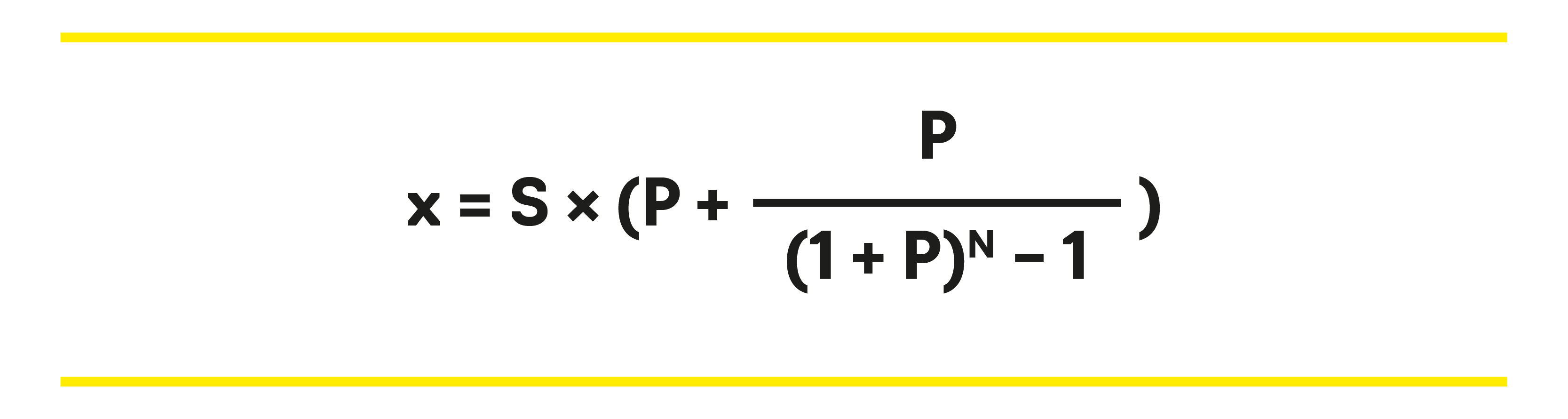
* «Рассчитать»
* «Сброс»

В поле «Сумма кредита» пользователь вводит количество денег, которые планируется взять взаймы

В поле «Ставка в процентах» вводится значение равное годовой процентной ставке банка в котором берется займ.

В поле «Количество лет для выплаты кредита» вводится ожидаемый заемщиком временной отрезок, за который он ожидает выплатить взятый кредит

Далее, после нажатия кнопки «Рассчитать» - рассчитывается размер ежемесячного платежа по аннуитетному принципу с помощью формулы:



Где:

X – Размер ежемесячного платежа

P – 1/100 доля процентной ставки (в месяц)

N – срок кредитования (в месяцах)

И сходя из полученного значения вычисляются так же общая сумма выплат и переплата по кредиту которые, в свою очередь, выводятся в графы соответственно:

* «Сумма платежа»
* «Переплата»

Размер ежемесячного платежа выводится в графе:

* «Сумма выплат в месяц»

Стоит упомянуть о кнопке «Сброс» которая очищает все поля ввода и выставляет в поля вывода значение 0.

**Препятствия и победы:**

Главная трудность, стоявшая передо мной, существовала в виде проблемы обработки всех видов исключений, которые можно ввести с клавиатуры.

Были предусмотрены меры об исключении ввода всевозможных знаков, букв, а также отрицательных чисел.

По мимо того, что все вышеперечисленное было реализовано - удалось создать ограничение на ввод значений в поля ввода:

* В поле «Сумма кредита» нет возможности ввести сумму более 999999999 (девятьсот девяносто девять миллионов девятьсот девяносто девяти тысяч девятисот девяноста девяти) т.к. ни один банк не согласиться кредитовать среднего заемщика на столь значительные суммы денег. Конечно если программа будет использоваться более крупными клиентами в виде мировых банков, которые кредитуют огромные корпорации, оперируя миллиардами – определенно ограничение будет снято, но в нынешний момент как автор, я оставляю это ограничение действующим.
* В поле «Ставка в процентах» невозможно ввести сумму более 999 (девятисот девяноста девяти) т.к. по закону Российской Федерации нельзя кредитовать заемщиков по ставке более 1 (одного) процента в день или 365 (трехсот шестидесяти пяти) процентов в год. Но для непреднамеренных ситуация и возможности использовать программу иностранными пользователями создал ограничение в 999 (девятьсот девяносто девять) процентов в год, что вызовет положительный отклик у пользователей иностранных банков.
* В поле «Лет для выплаты» установлено ограничение в 99 (девяносто девять) лет для выплаты кредита по причине того, что среднестатистический банк не возьмет на себя риск кредитовать заёмщик на срок более долгий. (Опять же при негативной обратной связи от общественности ограничение может быть снято)

**Визуализация задачи Комивояжера**

**Дизайн:**

Дизайн приложения прост и лаконичен. Существует белое поле, представляющее собой общее место, где будут располагаться нашли города.

Цвета элементов ввиду зеленых путей между городами, оранжевых кнопок «Matrix» и «Result», а также красной кнопки «Clear» так же основаны на триаде цветового треугольника, хоть и из-за малого количества цветных элементов это не так радикально отражается на визуально восприятии программы.

Больше проявляют себя пропорции и отношения элементов между собой. Они нестандартны, вытянуты, радикально различны по габаритам, но из за верных соотношений построенных на числах Фибоначчи - выглядит свежо и интересно.

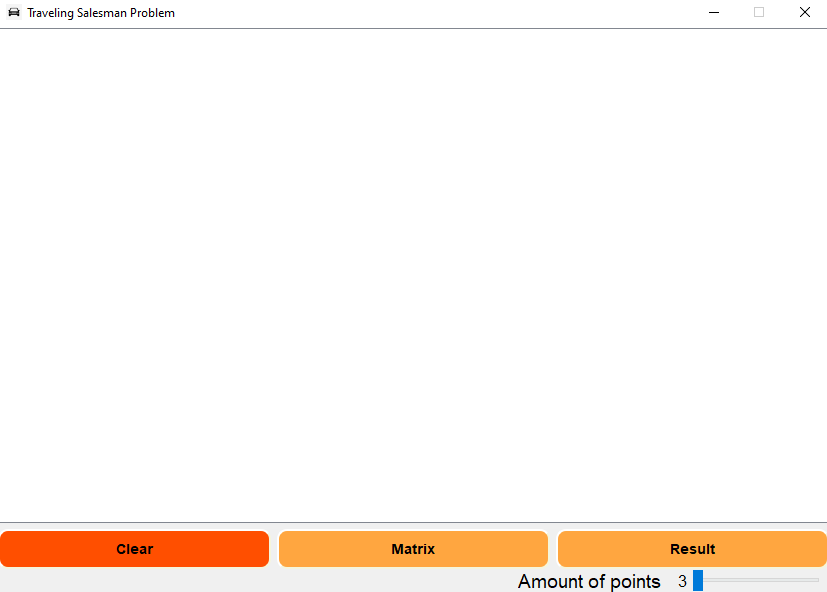


Рисунок 2: Начальное окно программы

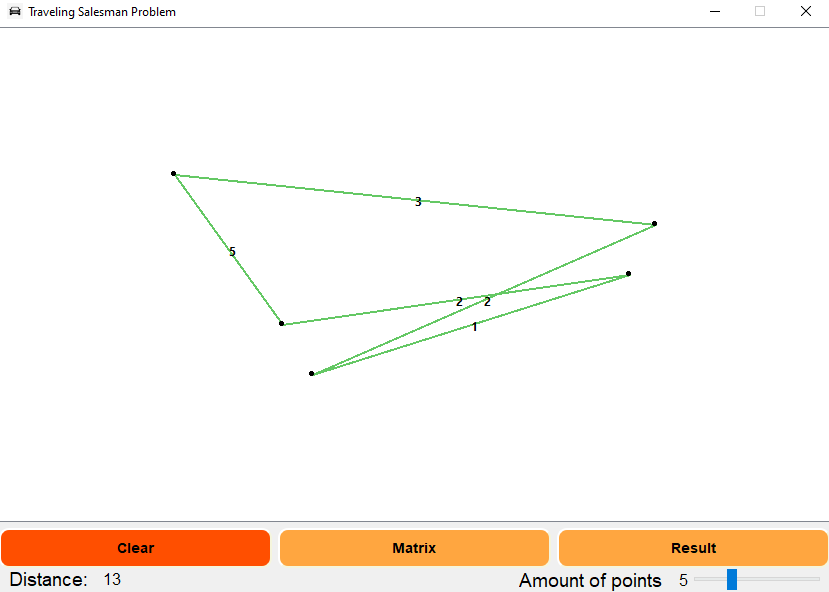


Рисунок 3: Результат выполнения при 5и городах

**Функционал:**

По мимо пространства для построения карты условных городов, на экране присутствуют слайдер для установки количества городов и 3 (три) кнопки действий:

* «Clear»
* «Matrix»
* «Result»

Слайдер создает на белом экране невидимые точки, условные города, и с каждым «тиком» слайдера создается новый условный город и общность этих условных городов меняет свои позиции, чем создают каждый раз новое положение точек и «свежую» картинку.

Кнопка «Clear» очищает сцену от полученного ранее результата программы.

Кнопка «Matrix» представляет собой функцию вывода матрицы смежности, в которую указываются расстояния между городами среди которых будет вестись подсчет наикротчайшего пути.

Кнопка «Result» отображает невидимые города и создает между ними зеленую линию, которая представляет собой путь из одного города в другой. Так же визуально представляется и число равное длине пути меж городами. После происходит поиск кротчайшего маршрута при условии прохождения всех городов, который выводится в появившейся снизу слева графе «Distance».

**Препятствия и победы:**

Наибольшую трудность собой представало создание идеи, как возможно реализовать создание городов на плоскости, не утруждая пользователя монотонно проставлять каждый город, и так же не перегрузить интерфейс приложения малоиспользуемыми функциональными кнопками.

Решение было найдено в концепции создания невидимых линий, на которых формируется невидимая точка и изменяющаяся при каждом движении слайдера пользователем. Тем самым интерфейс сохранил свою простоту, а пользователь не утруждается наблюдать функциональную работу программы.

Конечно же большую и приятную трудность составляло поиск решения проблем реализации возможностей платформы QT, таких как например полная реформация рабочего поля приложения, и создание невидимых элементов интерфейса, таких как поле «Distance».

Осуществление творческого задания дало мне абсолютно новый опыт разработки и сильно подтолкнуло как разработчика в профильных и прикладных навыках в специальности. Безмерно рад всему проделанному пути и результату, полученному в итоге .