

Требования к проекту

1 Введение

Анализатор валют – Программа должна содержать анализ по курсу валюты, проанализировать тенденции к росту или падению и наглядно показать результаты в виде графиков и схем. Программа не сможет выдавать точные сведения о том какой курс валюты будет завтра. Это должно быть простое и удобное приложение, в котором также будет встроен конвертер и можно будет не открывая калькулятор осуществлять простые операции.

2 Требования пользователя

2.1 Программные интерфейсы

ОС: Windows, Linux, Mac.

Язык программирования: Java FX.

Внешние библиотеки и сервисы: api Yahoo Finance (<http://finance.yahoo.com/>).

2.2 Интерфейс пользователя

Взаимодействие с пользователем в основном будет сопровождаться различными меню. Пользователю будет предоставляться выбор определенных валют, курс которых он хочет отслеживать. Также в приложении будут присутствовать иллюстрации в виде графиков и схем, чтобы пользователь смог отследить изменения курса валюты.

2.3 Характеристики пользователей

Данная программа будет полезна для людей, специализирующихся на анализе валютного рынка и занимающихся мониторингом валютно-обменных операций.

Основные характеристики потребителей: мужчины в возрасте 27-40 лет с высшим образованием, уровень дохода выше среднего, уверенный пользователь ПК.

2.4 Предположения и зависимости

Программа полностью зависит от сервера yahoo. В случае DDOS атак, механического повреждения сервера и других возможных причин выхода из строя сервера работа приложения осуществляться не будет.

3 Системные требования

Так как данное приложение будет реализовано на языке Java FX, который является кроссплатформенным, оно будет работоспособным на любой ОС. Также устройство, с которого будет запускаться приложение, обязательно должно иметь доступ в интернет.

3.1 Функциональные требования

1. Проектирование пользовательского интерфейса.
2. Проектирование методов взаимодействия с внешним сервисом.
3. Проектирование способа хранения пользовательской информации.
4. Разработка и тестирование вышеперечисленных компонентов.
5. Объединение разработанных компонентов в единую систему.

3.2 Нефункциональные требования

1. Простота пользовательского интерфейса (Интерфейс приложения должен быть максимально упрощен, чтобы пользователю было легче, приятней, не теряя лишнего времени, использовать данный продукт).
2. Надёжность (Данное приложение предназначено для работы с финансами, а значит из-за малейшего сбоя в системе пользователь может потерять большие деньги. Поэтому надёжность в данном приложении один из важных требований, её необходимо сделать максимальной, на сколько это возможно, иначе данное приложение не будет пользоваться спросом).
3. Расширяемость (В мире существует около 200 видов валют, а, значит, необходимо предусмотреть возможность интеграции более локализованных сервисов, которые смогут предоставить более редкие курсы валют, если этого пожелает пользователь).