**Техническое задание.**

Техническое задание на разработку аналитической программы “SteamCardsFarmer”.

1. **Общие сведения.**

Наименование: «SteamCardsFarmer».

Краткое наименование: программа.

Основания для проведения работ: задание кафедры МОСИТ.

Заказчик: кафедра МОСИТ.

Разработчик: ОАО Pomoika Inc.

Срок начала работ: 1 сентября 2018 года.

Срок окончания работ: 28 декабря 2018 года.

Программа разрабатывается в течение намеченного срока, в его конце сдается руководителю компании и выпускается на рынок.

1. **Назначение и цели создания системы.**
   1. **Назначение системы.**

Автоматизация процесса поиска игр с наличием атрибута «Коллекционные карты» в БД игровой платформы Steam.

* 1. **Цели создания системы.**

Создать программный продукт, способный находить и выдавать сущности – «Игры» (вместе с их основными параметрами) – с наличием атрибута «Коллекционные карты» в БД (Магазине) игровой платформы Steam, что позволит определить предполагаемый шанс окупаемости игры на основе ее стоимости и стоимости Коллекционных карточек, связанных с ней.

* 1. **Актуальность создания системы.**

Актуальность создания системы заключается в извлечении прибыли путем законных махинаций с ценами и Коллекционными картами игр, расположенных на платформе Steam, в условиях экономического кризиса в стране и поиска методов дополнительного заработка.

1. **Требования к системе.**
   1. **Требования к системе в целом.**

Система должна находить и выдавать упомянутые ранее сущности (Игры) с их основными параметрами и наличием атрибута «Коллекционные карты» в БД игровой платформы Steam. Эти сущности должны храниться в локальной БД, которая используется для подсчета вероятности получения прибыли из игры, или хотя бы ее окупаемости, после продажи Коллекционных карт.

* 1. **Требования к функциям, выполняемым системой.**

Функция поиска игр. Поиск должен игнорировать игры без целевого атрибута, а также бесплатные игры. Должна быть опциональная возможность ограничения поиска (максимальная цена, максимальное количество).

Функция прогнозирования окупаемости. Прогноз должен учитывать количество получаемых из одной игры карточек с учетом их уникальности. Также функция должна работать по более совершенному алгоритму, чем простой перебор.

Кроме того, крайне важно, чтобы в программе присутствовал графический интерфейс. Он должен быть интуитивно понятен, и при этом позволять выполнять все необходимые операции при должном уровне минимализма, а также иметь структуру, позволяющую при необходимости с минимальными затратами внести в него изменения.

* 1. **Требования к видам обеспечения.**

*Математическое обеспечение.* Постановка задачи: у каждой игры есть н карт (от 5 до 15) из которых случайным образом с возможными повторениями выпадает ровно половина с округлением в большую сторону. У каждой карты есть стоимость. Разработать эффективный алгоритм (не перебор "в лоб"), который будет подсчитывать (возможно, приблизительно) шанс того, что общая стоимость выпавших карт минус некоторый процент (константа) превысит стоимость игры, которой принадлежат эти карточки.

*Информационное обеспечение.* Должна присутствовать область временного хранения данных; информационный обмен существует между API программы и моделью MVVM; сбор данных в системе осуществляется с помощью специальной программы для обработки HTML-страниц; при разработке программного продукта должен использоваться язык C#, для создания графического интерфейса – язык разметки XAML; программный продукт должен работать на базе ОС Windows 10; разработка должна вестись в IDE Microsoft Visual Studio 2017.

1. **Подсистемы системы.**

Подсистемами данной системы являются:

* API для магазина Steam.
* База данных, хранящая в себе игры и необходимую информацию о них.
* Графический пользовательский интерфейс.

*API для магазина Steam.*

Данная подсистема представляет собой некий API для магазина Steam, который работает с html-файлами для поиска игр.

*База данных, хранящая в себе игры и необходимую информацию о них.*

Данная подсистема хранит в себе все необходимые для работы программы данные об играх, такие как: название, изображения, цена, коллекционные карты, и т. д.

*Графический пользовательский интерфейс.*

Данная подсистема обрабатывает информацию, поступающую от базы данных и API, и затем выводит ее в удобном для пользователя виде.

1. **Состав и содержание работ по созданию системы.**

Работы выполняются в три этапа:

* Проектирование системы (1 месяц)
* Разработка системы (3 месяца).
* Тестирование и ввод в эксплуатацию (1 месяц).

1. **Порядок контроля и приемки системы.**

Контроль и приемка программного продукта выполняются в два этапа:

* Тестирование программы тестировщиком.
* Приемка программы руководителем компании.

1. **Требования к документированию.**

Документирование должно производиться как в исходном коде, так и в специальных файлах. Для документирования следует использовать XML-комментарии, поддерживаемые IDE Microsoft Visual Studio 2017. Документирование тестовых прогонов программы не требуется.

1. **Источники разработки.**

Источником разработки является заказ разработки программного продукта кафедрой МОСИТ.