

### Задание 3. Уровень сложности – трудный (Hard Questions)

Мама готовит пирог на день рождения своему сыну. Насколько больше продукта X нужно взять больше, чем продукта Y?

(1)  $X = 10$

(2) Маме нужны вещества W, X, Y и Z в соотношении 15:5:2:1 и  $Y=4$

При проверке утверждений мы видим, что утверждения 2 достаточно для ответа на вопрос, а утверждения 1 недостаточно Ответ: (B)

**Заключение.** Опыт проведения тестирования GMAT показывает, что обучающиеся мало задумываются над анализом данных, чаще применяют стандартный алгоритм действий, поэтому задания на достаточность данных имеют невысокий коэффициент решаемости. Мы считаем, что такие задания целесообразно разрабатывать и использовать в образовательном процессе в школе и в вузе, они отличаются от традиционных заданий, развивают аналитические способности, создают условия для поиска нестандартных подходов к решению задачи. Создание банка таких заданий позволит студентам самостоятельно проводить оценивание способности анализировать достаточность данных [1], достигнутые результаты студенты смогут размещать на странице учебных достижений в электронном портфолио [2], задания такого типа целесообразно включать в фонд оценочных средств по дисциплине [3].

1. Медведева И.Н., Мартынюк О.И., Панькова С.В., Соловьева И.О. Самооценка сформированности компетенций студентов первого курса физико-математического факультета в условиях реализации ФГОС ВПО // Вестник Псковского государственного университета. Серия: Естественные и физико-математические науки. – 2012. – № 1. – С. 129–146.
2. Медведева И.Н., Мартынюк О.И., Панькова С.В., Соловьева И.О. К вопросу о формировании электронного портфолио обучающегося // Вестник Псковского государственного университета. Серия: Естественные и физико-математические науки. – 2014. – № 5. – С. 134–140.
3. Медведева И.Н., Мартынюк О.И., Панькова С.В., Соловьева И.О. О подходах к формированию фонда оценочных средств по образовательной программе // Вестник Псковского государственного университета. Серия: Естественные и физико-математические науки. – 2015. – № 7. – С. 75–91.
4. <https://ru.scribd.com/document/72776347/The-Official-Guide-for-GMAT-Review-12th-Edition-Part1-3> (официальное пособие для подготовки к GMAT)
5. <https://ru.scribd.com/doc/96562925/Kaplan-GMAT-800-Second-Math-Part>

## СЕРВИС СБОРА ИНФОРМАЦИИ О АППАРАТНОМ И ПРОГРАММНОМ ОБЕСПЕЧЕНИИ КОМПЬЮТЕРОВ И УДАЛЕННОЙ УСТАНОВКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

*Цепота П.С., Шульга Р.Д., Лавренов А.А.,*

*студенты ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь*

*Научный руководитель – Новый В.В.*

При работе с компьютером пользователи сталкиваются с проблемой отслеживания конфигурации компьютера и актуальных версий ПО. Данная проблема становится наиболее актуальной, когда необходимо иметь данные о большом количестве компьютеров. Например, в университете или организации, в которой активно используются компьютеры. На данный момент существуют программы, которые могут предоставить пользователю конфигурации его машины, и/или установленное ПО. Однако, как правило, они не способны передавать собранные данные на удалённый сервер.

Целью нашей работы является создание полноценной системы для сбора информации об аппаратной части компьютера, установленном на нем программном обеспечении, а также позволяющей выполнять фоновую установку программ, загружаемых с сервера.

**Материал и методы.** Использовалась система контроля версий GitHub. Применялся подход Agile цикла разработки программного обеспечения. При разработке применялись методы объектно-ориентированного программирования и проектирования.

**Результаты и их обсуждение.** В результате была разработана система, состоящая из 3 частей. Первая – это сервис(служба) Windows который работает на машине и выполняет сбор информации и установку ПО. Вторая – сервер для централизованной обработки, хранения и предоставления данных. Третья – графический веб-клиент для взаимодействия с функционалом сервера. Распределённость системы на несколько модулей значительно упрощает процесс разработки.

Сервис был написан на платформе .NET Framework. [1] Он разбит на несколько модулей:

- Hardware отвечает за сбор информации о комплектующих компьютера.
  - HttpServer отвечает за сетевое общение сервиса с сервером и выполняет вызов других модулей приложения.
  - Network отвечает за фильтрация ip адресов и непосредственный запуск http сервера.
  - Service отвечает за установку и запуск сервиса(службы).
  - Software отвечает за сбор информации о программах и их установку.
- Графический веб-клиент был написан на Angular 6. Он разбит на несколько директорий:
- Директория Audiences, компоненты которой отвечают за отображение
  - главной страницы приложения, на которой отображаются доступные аудитории.

- Директория Computers, компоненты которой отвечают за отображение доступных компьютеров, их аппаратного и программного обеспечения, а также за добавления новых компьютеров.
- Директория Files, компоненты которой отвечают за отображение файлов, хранящихся на сервере и за добавление новых файлов.
- Директория Requests, компоненты которой отвечают за отображение запросов, хранящихся на сервере и за добавление новых запросов.
- Директория Users, компоненты которой отвечают за отображения пользователей, зарегистрировавшихся в системе, а также за регистрация новых пользователей и авторизацию.
- Директория NavBar, компоненты которой отвечают за навигацию в приложении.

Сервер для централизованной обработки представляет собой Spring Boot[2] приложение с базой данных, взаимодействующее с веб-клиентом и сервисами на компьютерах.

Архитектура приложения имеет следующие директории:

- controllers – хранит классы обрабатывающие пользовательские запросы.
- service – основной модуль приложения, обрабатывающий данные в соответствии с бизнес логикой приложения.

- entity – хранит описание сущностей, хранящихся в базе данных.

- repository – обеспечивает доступ к базе данных.

В приложении используются такие расширения как:

- Google Guava – упрощает описание сущностей и имеет большое количество средств для работ с коллекциями
- Lombok – предоставляет функционал способствующий уменьшению количества и читабельности кода.

**Заключение.** Была спроектирована, разработана и протестирована система позволяющая упростить отслеживание актуального состояния программного и аппаратного обеспечения. Система является полезным инструментом администрирования большого количества компьютеров.

1. Сервисы Windows [Электронный ресурс] – 2019. – Режим доступа: <https://docs.microsoft.com/en-us/dotnet/framework/windows-services/>. – Дата доступа: 25.02.2019
2. Документация по Spring Boot [Электронный ресурс] – 2019. – Режим доступа: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>. – Дата доступа: 26.02.2019

## ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ РАСТЕНИЙ ПО ФОТОГРАФИЯМ РОСТКОВ

**Шульгин И.В.,**

*студент 4 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь*

Научный руководитель – Кухарев А.В., канд. физ.-мат. наук

Необходимость автоматического определения вида растения по фотографиям его ростков возникает в сельском хозяйстве в связи с необходимостью обнаружения сорняков и составления их карт [1].

За последние десятилетия был разработан ряд архитектур глубоких сверточных нейронных сетей, таких как Inception (GoogleNet) и ResNet [2], предназначенных для эффективного распознавания образов.

Цель настоящей работы – установить, какие архитектуры глубоких нейронных сетей пригодны для распознавания растений по фотографиям их ростков и, в частности, для выявления среди них сорняков.

**Материал и методы.** Для обучения нейронной сети использовался набор данных, размещенный в открытом доступе [3]. Он включает в себя 4750 фотографий ростков 12 видов растений. Примеры изображений показаны на рисунке 1. Для обучений нейронной сети использовалось 70% изображений, оставшаяся часть – для тестирования сети.

Эксперимент проводился для нескольких нейросетевых моделей: ResNet с разным количеством слоёв (18, 34, 50, 101 и 152), AlexNet, VGG19, Densenet121 и Inception-v3.

Обучение нейросетевой модели осуществлялось по методу «переноса обучения» (transfer learning), который заключается в использовании весов нейросетевых моделей, обученных ранее на других изображениях. В нашем случае использовался набор данных ImageNet. В качестве оптимизатора использовался Adagrad с коэффициентом обучения 0,0005.

**Результаты и их обсуждение.** В таблице 1 приведены значения точности классификации вида растения из тестовой выборки различными нейросетевыми моделями после 20 эпох обучения. В данном случае все модели являлись предобученными на данных ImageNet, а оптимизация проводилась для всех слоёв сети. Наилучший результат показала модель ResNet101 – она дала 97,78% правильных ответов на тестовом наборе.