

Для улучшения работы по борьбе с наркотиками предложено создание общей для всех ведомств информационной системы, предоставляющей широкий выбор сервисов для обмена и анализа полученных данных и своевременного выявления и противодействия уже существующим, а также появляющимся тенденциям наркобизнеса. Данная система должна соответствовать ряду требований. В ней должно быть достаточно полной и достоверной информации об уже исследованных и введенных под контроль веществах, кроме того, данные должны постоянно и своевременно обновляться. В систему должен быть включен ряд сервисов для анализа и обработки данных, например, быстрая система поиска по различным параметрам, анализатор химической структуры вещества, подструктур и производных веществ. Программный продукт данного рода должен представлять собой единую информационную сеть ведомств как внутри одного государства, так и на международной основе. Система должна быть отказоустойчивой, простой и удобной в использовании, оптимизированной для работы в любых условиях. Кроме того, данный программный продукт должен соответствовать всем требованиям информационной безопасности, новейшими системами защиты от взлома и атак посредством вредоносных программ.

**Заключение.** Таким образом, на основе анализа был выявлен ряд проблем в сфере борьбы с наркотиками. Используемые сейчас методы борьбы можно назвать устаревшими в силу существенных изменений в тенденциях развития наркобизнеса. Важную роль играет отсутствие используемых в борьбе с наркотиками информационных технологий, которые могли бы оптимизировать и ускорить работу всех причастных к данной работе ведомств, позволяя оперативно и своевременно реагировать на любые изменения в наркобизнесе.

1. АИПСИН [Электронный ресурс]. – Режим доступа: <https://aipsin.com/> – Дата доступа: 15.03.2023.

## **ПРИМЕНЕНИЕ ФУНКЦИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ В TEST AUTOMATION FRAMEWORK (TAF) В ТЕСТИРОВАНИИ API**

*Грицкевич Н.С., Китаров Д.А.,*

*студенты 3 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь*

*Научный руководитель – Новый В.В., ст. преподаватель*

Современное программное обеспечение в настоящее время делится на две части UI и API. UI отвечает за сохранение или показ данных, отображение анимации и использование приятной цветовой гаммы на элементах страницы. API – отвечает за преобразования информации и обработку в зависимости цели проекта, что делает её важной составляющей при работе ПО, так как UI зависит от API (если API перестанет работать, то UI не сможет ничего отображать, а обратное утверждение неверно). В данной статье мы рассмотрим, как с помощью функционального программирования [1] создать классы для гибкого и простого использования для автоматизации тестирования [2] API части, что позволит получить прирост производительности и уменьшит количество дублирующего кода.

**Материал и методы.** Материалом исследования является возможность использования функционального программирования языка java для создания “обёрток” над API response [3].

**Методы исследования** – анализ, методы объектно-ориентированного и функционального программирования на языке java.

**Результаты и их обсуждение.** При рассмотрении пирамиды тестирования [4] мы можем заметить, что API слой находится в середине пирамиды и, следовательно, покрывается инженерами автоматизаторами большим количеством тестовых методов. В основном при проверке response модели нам требуется проверять возвращаемый

статус код, а также в зависимости от тест-кейса тело ответа или поле присущее этой модели.

В данном случае, основываясь на “best practice” написания тестовых методов, мы имеем на каждый запрос как минимум 2 тестовых метода: проверка статус кода, тела запроса или отдельных их полей. Далее будет предложена обёртка над response моделью для сокращения количества тестов почти в 2 раза. Но основываясь на “best practice”, нам так же желательно отделить кодовую логику от тестов. Для этого мы дополнительно опишем два класса основанных на функциональном программировании и так называемом “Fluent API”, что позволит получать необходимые данные из тела запроса в независимости от типа данных и вида данных (коллекция или класс).

Далее будет приведён пример данных классов в виде скриншотов с назначением этих классов:

```
@Getter
@EqualsAndHashCode
@AllArgsConstructor(access = AccessLevel.PRIVATE)
public class ResponseModel<T> {
    private Integer statusCode;
    1 usage
    private T body;
    3 usages
    public static <T> ResponseModel<T> of(Integer statusCode, T model) {return new ResponseModel<>(statusCode, model);}
    3 usages
    public static <T> ResponseModel<T> of(Response response, Class<T> zClass) {return new ResponseModel<>(response.getStatusCode(), response.as(zClass));}
    7 usages
    public Operation<T> operation() { return Operation.from(this.body); }
}
```

```
@AllArgsConstructor(access = AccessLevel.PRIVATE)
public class CollectionOperation<T> {
    3 usages
    private Stream<T> value;
    1 usage
    public static <T> CollectionOperation<T> from(Collection<T> value) {return from(value.stream());}
    3 usages
    private static <T> CollectionOperation<T> from(Stream<T> value) {return new CollectionOperation<>(value);}
    public <U> CollectionOperation<U> map(Function<T, U> function) {return from(value.map(function));}
    public CollectionOperation<T> filter(Predicate<? super T> function) {return from(value.filter(function));}
    public <R, A> R get(Collector<? super T, A, R> collector) {return value.collect(collector);}
}
```

```
@AllArgsConstructor(access = AccessLevel.PRIVATE)
public class Operation<T> {
    3 usages
    private T value;
    2 usages
    public static <T> Operation<T> from(T value) {return new Operation<>(value);}
    public <U> Operation<U> map(Function<T, U> function) { return from(function.apply(value)); }
    public <U> CollectionOperation<U> flatMap(Function<T, Collection<U>> function) {return CollectionOperation.from(function.apply(value));}
    public T get() { return value; }
}
```

Класс ‘ResponseModel’ используется в виде контейнера самого тела запроса и статус кода из API ответа, благодаря этому и достигается уменьшение количества тестовых методов. Однако бывают случаи, что нам приходится проверять определенное поле или множество полей. Для этого мы можем использовать классы ‘Operation’ и ‘CollectionOperation’, первый - используется для преобразования классов, а второй для работы над коллекциями элементов. Использование ‘generic’ классов [5] и функциональных интерфейсов позволяет абстрагироваться от привязки к типам данных и методов.

**Заключение.** В работе рассмотрены наборы классов для тестирования API в TAF на java. Результаты данной работы позволяют использовать разработанные классы в автоматизации API тестов, как руководство для начинающих инженеров по автоматизации тестирования и в дисциплинах, относящихся к этой тематике.

1. Functional Programming in Java: How functional techniques improve your Java programs 1st Edition/ Pierre-Yves Saumont – 2017. – 472 с.

2. Тестирование и оценка качества программного обеспечения [Электронный ресурс]: [учеб.-метод. комплекс] для студентов, обучающихся по спец. 1-31 03 07 Прикладная информатика / М-во образования Республики Беларусь, Учреждение образования "Витебский государственный университет имени П.М. Машерова", Математический фак., Каф. информатики и информационных технологий. – Электрон. текстовые дан. (1 файл: 11 Мб). – Витебск, 2014. – URL: <https://lib.vsu.by/uchmat/?download=139417> (дата обращения 01.03.2023).

3. Rest-assured wiki [Электронный ресурс]. – Mode of access: <https://github.com/rest-assured/rest-assured/wiki/Usage#getting-response-data>. – Data of access: 21.03.2023.

4. Подробнее про пирамиду тестирования [Электронный ресурс]. – Mode of access: <https://habr.com/ru/post/672484>. – Data of access: 21.03.2023.

5. Java Generics and Collections: Speed Up the Java Development Process/ Maurice Naftalin, Philip Wadler – 2009. – 285 с.

## РАЗРАБОТКА ПРОФОРИЕНТАЦИОННОГО VR-ПРИЛОЖЕНИЯ

*Грищенко М.В.,*

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

*Научный руководитель – Мехович А.П., канд. физ.-мат. наук*

Виртуальная реальность – технология, позволяющая перенестись в мир компьютерной симуляции. Благодаря очкам и контролерам виртуальной реальности можно в полной мере взаимодействовать с этим миром. Сегодня VR нашел применение в различных сферах жизнедеятельности человека таких как производство, медицина, развлечения, образование, наука, культура и др.

Важным шагом в жизни любого человека является выбор профессии. В 2023 году рынок насчитывает более миллиона VR-приложений. Лишь небольшая часть из них относится к профориентационным.

Целью данной работы является создание профориентационного VR-приложения, которое демонстрирует направления специальностей факультета математики и информационных технологий Витебского государственного университета имени П.М. Машерова.

**Материал и методы.** Реализация VR-приложения выполнялось средствами и методами создания 3D-графики и компьютерный игр: Blender и Unity.

**Результаты и их обсуждения.** Визуализация и геймификация – самые понятные и популярные инструменты для молодёжи. Именно эти два фактора объединяет профориентация в виртуальной реальности.

Результатом исследования является разработка профориентационного VR-приложения. С помощью данного приложения можно познакомиться с ключевыми направлениями специальностей факультета математики и информационных технологий Витебского государственного университета имени П.М. Машерова: математика, физика, экономика и программирование.

Данное VR-приложение представляет собой трёхмерный образовательный и познавательный квест по следующим станциям:

- Math Saber – пользователю необходимо разубить мечом летящий на него куб с числом, являющимся верным ответом на поставленный арифметический вопрос;
- Try to relocate – головоломка, основанная на физических законах, в которой необходимо переместить шар из точки А в точку Б, используя различные объекты, меняющие траекторию движения шара;
- My biasness startup – пользователю необходимо выполнить наиболее выгодное вложение денежных средств;