



## ТЕХНИЧЕСКИЕ НАУКИ

УДК 004.942

### Разработка архитектуры системы управления комплексом виртуальных лабораторных стендов по защите от утечек по техническим каналам

*Д.А. Короченцев, А.А. Нардид*

Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)

**Аннотация.** Проведен анализ информации о технологиях, подходящих для разработки системы виртуальных стендов. По его результатам была выбрана наиболее приемлемая технология для создания такой системы. Проведено проектирование архитектуры управления комплексом виртуальных лабораторных стендов в части защиты от возможных утечек по техническим каналам. Результаты и выводы данного исследования могут быть полезными при разработке и внедрении подобных систем в образовательных и научных учреждениях.

**Ключевые слова:** информационная безопасность, подготовка специалистов, программно-аппаратные стенды, виртуальные стенды, техническая защита информации, Unity Engine

### Development of the control system architecture for the complex of virtual laboratory stands for protection against leaks through technical channels

*Denis A. Korochentsev, Aleksandr A. Nardid*

Don State Technical University (Rostov-on-Don, Russian Federation)

**Abstract.** The article analyzes the information on technologies suitable for the development of the virtual stand system. According to the results of the analysis the most suitable technology for the system development was selected. The design of architecture of virtual laboratory stand complex control system in terms of protection from possible leaks through technical channels was carried out. The results and conclusions of this study can be useful for the development and implementation of similar systems in educational and scientific institutions.

**Keywords:** information security, training of specialists, software and hardware stands, virtual stands, technical protection of information, Unity Engine

**Введение.** Техническая защита информации приобретает все большее значение в последние годы. Это обусловлено прежде всего активным развитием методов и средств добывания информации, позволяющих несанкционированно получать большие ее объемы на безопасном расстоянии, оснащением служебных, жилых помещений, а в последнее время и автомобилей разнообразной электро- и радиоэлектронной аппаратурой, являющейся источником случайных опасных сигналов. Построение эффективной системы защиты информации возможно лишь при наличии высококлассных специалистов в области информационной безопасности, подготовка которых является одним из приоритетных направлений в РФ.

Таким образом, разработка системы управления комплексом виртуальных лабораторных стендов в настоящее время является в полной мере актуальной задачей. Предметом данного исследования являются методы, используемые при подготовке специалистов в области защиты информации от утечек по техническим каналам, а объектом — технические каналы утечки информации. Теоретическая значимость работы заключается в представлении анализа технических каналов утечки информации и методов подготовки специалистов в области защиты информации. Практическая же значимость заключается в разработке программного средства, реализующего систему управления комплексом виртуальных лабораторных стендов по идентификации каналов утечек информации по техническим каналам.

Целью исследования является анализ архитектуры системы управления комплексом виртуальных лабораторных стендов по защите от утечек по техническим каналам. Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ архитектуры разрабатываемой программы, составить схему классов и их взаимодействия;
- проанализировать программу, разработав подробный алгоритм её рабочего процесса;
- рассмотреть применяемые технологии для разработки системы управления комплексом виртуальных

лабораторных стендов по защите от утечек по техническим каналам.

**Основная часть.** Первоначально следует провести анализ стека технологий, используемого для разработки системы управления комплексом виртуальных лабораторных стендов по защите от утечек по техническим каналам. В рамках данной работы выбор его производился на основе требований, которые включают в себя:

- легковесность приложения, которая подразумевает возможность его запуска на обусловленной в ТЗ конфигурации компьютера;

- масштабируемость, т. е. возможность дальнейшего развития проекта.

В ходе анализа существующих на данный момент решений были рассмотрены следующие из них: Unreal Engine и C++, Unity Engine и C#, Godot Engine и GDScript, Windows Forms [1].

Unreal Engine — кроссплатформенный набор графических инструментов, позволяющий создавать преимущественно 3D-проекты с высокодетализированной графикой. При этом данный набор имеет хорошие показатели производительности. Основным сценарным языком для него является C++. Также он поддерживает встроенную систему визуального программирования Blueprints.

Unity Engine — гибкий кроссплатформенный набор графических инструментов, позволяющий создавать как 2D, так и 3D-проекты. Этот набор инструментов имеет высокую популярность среди начинающих разработчиков, что привело к тому, что на данный момент существует огромное количество обучающего материала [2].

Godot Engine open source — кроссплатформенный набор графических инструментов. Является самым молодым среди рассматриваемых в данной работе решений. Его главный недостаток заключается в том, что он ещё развивается, поэтому многие возможности или не доработаны до стабильного состояния, или отсутствуют вовсе. Также в данном наборе используется проприетарный сценарный язык GDScript.

Windows Forms — это библиотека графического интерфейса, включённая в состав Microsoft .NET Framework. Данная библиотека позволяет создавать событийно-ориентированные приложения. Основным достоинством данного варианта является то, что эта библиотека, как часть Microsoft .NET Framework, может быть использована с любым языком, поддерживающим данную платформу, в том числе с C#, C++, Visual Basic и др.

На основании предъявленных к программной реализации требований следует выбрать технологический стек, который обеспечивает легковесность приложения и возможность дальнейшего масштабирования проекта. Наиболее подходящим вариантом из представленных является Unity Engine и C#. Unity Engine позволяет создавать как 2D, так и 3D-проекты, имеет гибкие возможности для настройки и прототипирования, а также широкую поддержку и большое количество обучающих материалов. Кроме того, использование C# в Unity Engine позволяет создавать проекты с высокой производительностью и возможностью масштабирования. Для написания кода использовалась IDE Microsoft Visual Studio.

Некоторые недостатки, которые были отмечены в отношении Unity Engine, могут быть устранены с помощью соответствующих настроек и дополнительных инструментов.

Windows Forms, в свою очередь, может быть подходящим вариантом для разработки событийно-ориентированных приложений, но в данном случае Unity Engine и C# более подходят для создания игрового проекта с возможностью масштабирования в будущем.

Unreal Engine и Godot Engine хотя и являются мощными решениями, могут быть слишком тяжеловесными для данного проекта, а их сценарные языки (C++ и GDScript) — сложными для начинающих разработчиков. Кроме того, Godot Engine является более молодым проектом и может не обеспечивать нужный уровень стабильности и поддержки.

Ранее авторы статьи уже разработали лабораторный 3D-стенд моделирования утечек информации за счёт акустических и виброакустических каналов утечки информации, он позже может быть доработан до полноценной системы управления лабораторными стендами. Его компоненты разрабатывались на языке C# с использованием платформы .NET Framework 4.5.2 в среде разработки Unity3D 2020.3 и Microsoft Visual Studio 2022. В качестве паттерна проектирования использовался подход Model-View-Presenter, или MVP [3]. С помощью подхода MVP достигается более гибкая и расширяемая архитектура приложения.

Основные классы, из которых состоит разработанное программное средство:

- CamSwitch — содержит в себе пользовательский интерфейс, который отвечает за переход пользователя между этапами (сценами) выбранной лабораторной работы, например переход пользователя на этап выбора контрольных точек (КТ) или на этап расстановки контрольно-измерительного оборудования (КИО). Реализован с помощью объектов типа Camera и их функции SetActive в связке с использованием элементов пользовательского интерфейса;

- ControlPoint — реализует взаимодействие пользователя и объектов на сцене (экране) с помощью метода рендеринга ray casting, который заключается в том, что вся сцена строится на основе замеров пересечения лучей с визуализируемой поверхностью. В данном случае такой поверхностью является ограждающая конструкция, на

которой необходимо отметить контрольные точки для замеров [4];

- EmptyProps — пустые, невидимые объекты с триггерами внутри для определения правильности постановки того или иного контрольно-измерительного оборудования внутри сцены. Помимо этого, содержат свойства моделей расставляемого КИО, меняющиеся в зависимости от нажатых пользователем элементов интерфейса и позволяющие передвигать оборудование по сцене;

- Check — необходим для проверки всех условий успешного завершения лабораторной работы, т. е. сверки с помощью метода ray casting введенных пользователем значений на поздних этапах и выбранных точек установки КТ и КИО [5];

- KIOEvents — отвечают за логику собранного пользователем измерительного стенда и взаимодействие его с моделями расставленного оборудования [6]. Моделирует значения параметров используемых технических средств в соответствии с выбранной пользователем среднегеометрической частотой и вызывает интерфейс того или иного прибора по клику на него;

- UI — содержит пользовательский интерфейс, присущий тому или иному этапу, например вызываемый кликом интерфейс оборудования или итоговая таблица, куда пользователю необходимо внести итоговые значения расчетов и провести их проверку;

- Database — отвечает за передачу и обеспечение хранения в базе MySQL данных пользователей-студентов, сведений о выполнении ими лабораторных работ и оценок по ним;

- UserMark — формирует и выставляет оценки пользователям по итогам сдачи лабораторных работ. На протяжении всей работы ведется подсчет допущенных ошибок, исходя из их количества выставляется та или иная оценка [7].

Отдельно стоит отметить пространства имен System.IO и System.Xml.Serialization, с помощью которых по итогам проверки финального этапа лабораторной работы таблица с результатами конвертируется в отдельный файл формата Microsoft Excel. На рис. 1 изображена схема со всеми классами, соединенными между собой с помощью UI.

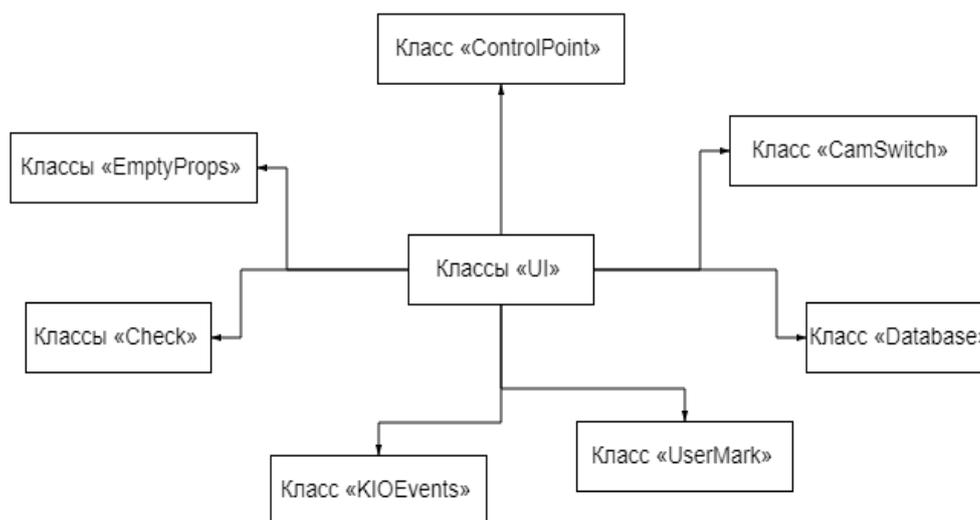


Рис. 1. Схема классов

Алгоритм работы с программой изображен на рис. 2, 3 и 4. Работа с системой начинается со входа пользователя в свой профиль и выбора лабораторной работы на выполнение.

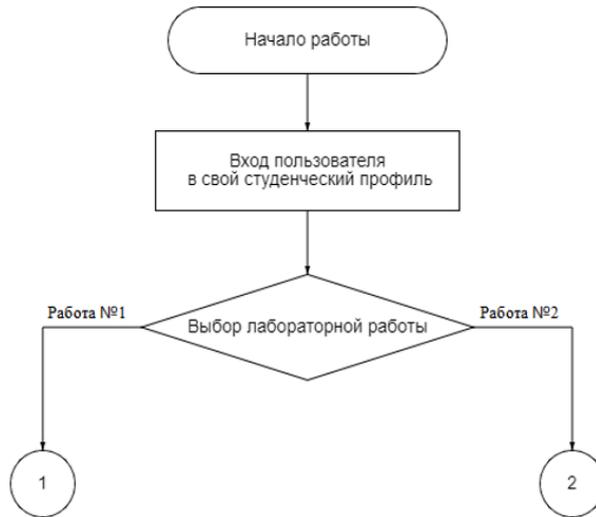


Рис. 2. Начало алгоритма

Далее осуществляется переход к этапу выполнения лабораторной работы. В первом случае будет начато выполнение работы по идентификации виброакустического канала утечки информации.

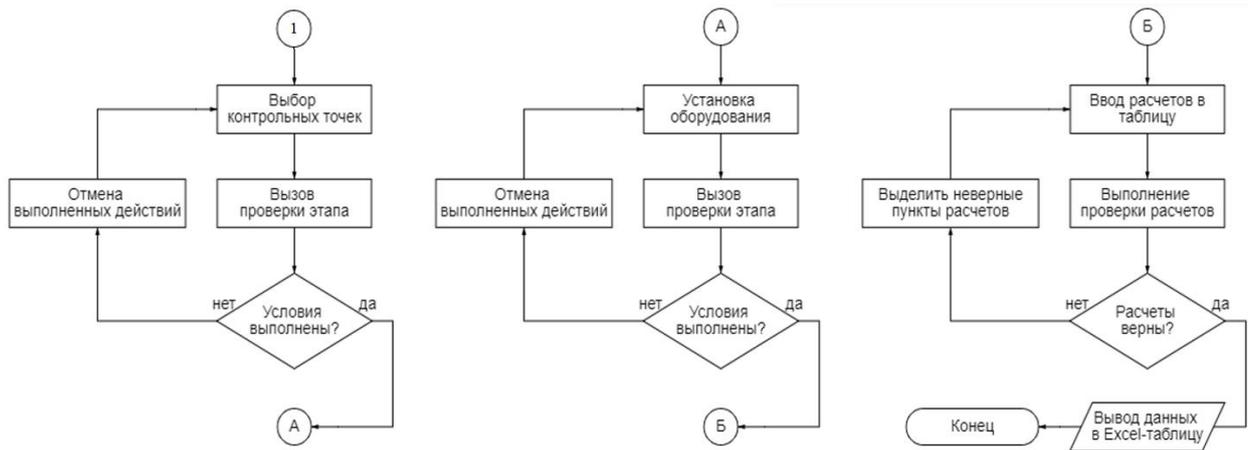


Рис. 3. Алгоритм выполнения первой лабораторной работы

При выборе второй лабораторной работы будет начато выполнение работы по идентификации утечки информации по каналу низкочастотных акустоэлектрических преобразований.

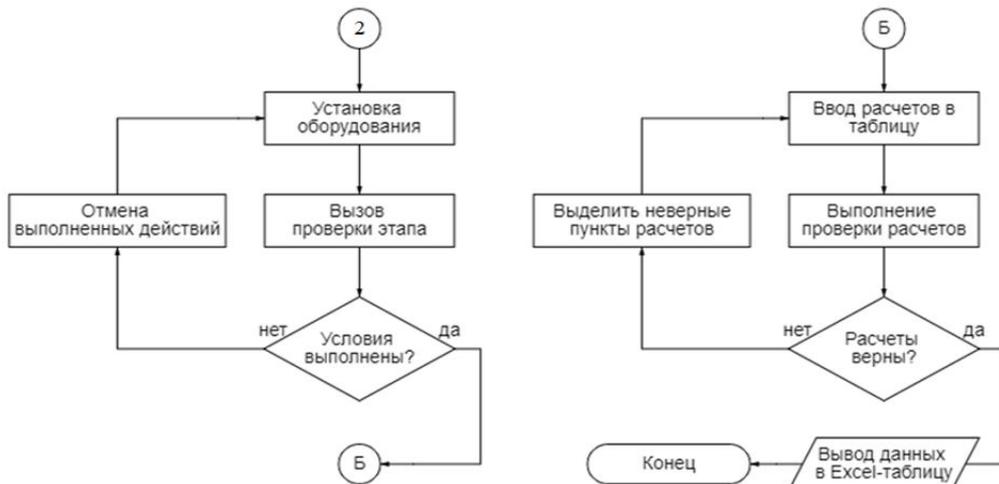


Рис. 4. Алгоритм выполнения второй лабораторной работы

Аналогичные алгоритмы с небольшими изменениями из-за разницы методик применяются и для других лабораторных работ, которые могут быть включены в разрабатываемую систему управления комплексом виртуальных лабораторных стендов в виде дополнительного модуля.

**Заключение.** В ходе данного исследования были подробно проанализированы различные технологии для создания системы управления комплексом виртуальных лабораторных стендов по защите от утечек по техническим каналам и были выбраны оптимальные и наиболее удобные для разработки — Unity Engine и Microsoft Visual Studio. Помимо этого, в ходе работы была предложена архитектура создаваемой системы управления лабораторными стендами, подробно описаны все классы и их взаимодействие между собой в проекте.

#### Список литературы

1. *Руководство Unity 2021.3 (LTS)*. UnityHub. URL: <https://unityhub.ru/manual/UnityManual> (дата обращения: 13.07.2023).
2. *Creating and Using Materials*. Unity. URL: <https://docs.unity3d.com/ru/530/Manual/Materials.html> (дата обращения: 13.07.2023).
3. Gamma E, Helm R, Johnson R, Vlissides J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley; 1995. 395 p.
4. Felicia P. *Unity From Zero to Proficiency (Foundations)*. Independently Published; 2019. 240 p.
5. *Мероприятия по выявлению технических каналов утечки информации. Оценка защищенности информации от утечки по ТКВИ*. ИНТУИТ. URL: <https://intuit.ru/studies/courses/2291/591/lecture/12707?page=2&ysclid=19> (дата обращения: 13.07.2023).
6. Волков Д.С., Козлов А.О. Методика оценки защищенности систем передачи данных от утечки речевой конфиденциальной информации по каналам электроакустических преобразований. *Научный поиск*. 2014;2(5):4–6.
7. Дураковский А.П., Куницын И.В., Лаврухин Ю.Н. *Контроль защищенности речевой информации в помещениях. Аттестационные испытания вспомогательных технических средств и систем по требованиям безопасности информации*. Москва: НИЯУ МИФИ; 2015. 152 с.

*Об авторах:*

**Короченцев Денис Александрович**, исполняющий обязанности заведующего кафедрой «Кибербезопасность информационных систем» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), кандидат технических наук, доцент, [ORCID](#), [center-bit@yandex.ru](mailto:center-bit@yandex.ru)

**Нардид Александр Александрович**, студент кафедры «Кибербезопасность информационных систем» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), [nardidalexander@yandex.ru](mailto:nardidalexander@yandex.ru)

*About the Authors:*

**Denis A. Korochentsev**, acting head of the Cybersecurity of Information Systems Department, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), [ORCID](#), Cand.Sci. (Eng.), associate professor, [center-bit@yandex.ru](mailto:center-bit@yandex.ru)

**Aleksandr A. Nardid**, student of the Cybersecurity of Information Systems Department, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), [nardidalexander@yandex.ru](mailto:nardidalexander@yandex.ru)