

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № «BikeStat: Разработка аналитического приложения с
персональными рекомендациями для велосипедистов»**

1. Условия

В настоящее время велосипедный спорт становится все более популярным видом активного отдыха, а велосипед входит в нашу жизнь как удобное и незаменимое средство передвижения, показавшее свою эффективность как в городской среде, так и за её пределами. Повышенный интерес к использованию этого вида транспорта мотивирует инженеров, занятых проектированием и разработкой решений в области современных информационных технологий, все больше и больше задумываться над вопросом усовершенствования информационного обеспечения, способного сделать процесс передвижения на велосипеде более удобным и эффективным. Современные программные комплексы призваны стать надежными помощниками тем велосипедистам, которые решили идти в ногу со временем и использовать информационные технологии в своих путешествиях. Такие программные комплексы, а проще говоря - приложения, могут быть полезны при составлении маршрутов передвижения, планировании поездок, составлении графика физической активности и рекомендаций с учётом индивидуальных особенностей и целей велосипедиста.

Вам предлагается разработать программный комплекс (программный продукт) для велосипедистов, который сможет осуществлять планирование и анализ поездок, отслеживать местоположения пользователя во время поездки, формировать набор персональных рекомендаций с учётом уровня подготовки пользователя, накапливать статистические данные физической активности, затраченных калорий и пройденного пути.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № «BikeStat: Разработка аналитического приложения с
персональными рекомендациями для велосипедистов»**

Целью разработки программного комплекса является повышение эффективности тренировочного процесса пользователя.

Для осуществления анализа поездок предлагается использовать GeoAPI для построения маршрутов и предоставляемое организаторами API, моделирующее работу носимых устройств, измеряющих пульс велосипедиста, для формирования персональных рекомендаций.

2. Техническое задание

Необходимо разработать программный продукт – приложение для тренировки велосипедистов. Приложение должно формировать персональные рекомендации о планировании поездок на основе собранной статистики: скорости передвижения по маршруту, пройденного пути и частоте сердечных сокращений. План поездки должен включать в себя: выбор маршрута, скорость передвижения, предполагаемое время в пути и дату поездки.

Функциональное задание:

Обязательная функциональность:

- построение и трекинг велосипедного маршрута;
- оценка (расчетная) пройденного маршрута на основании общих¹ данных:
 - легкий;
 - средний;
 - сложный;

¹ Общие данные остаются на усмотрение разработчика, но должны быть основаны на приближенных к реальным физическим значениям

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**

Профиль «Информационные технологии»

Командный кейс № «BikeStat: Разработка аналитического приложения с персональными рекомендациями для велосипедистов»

- после окончания маршрута получение от носимого устройства данных о пульсе (минимальный, максимальный, средний) за время поездки;
- корректировка расчетной оценки сложности пройденного маршрута на основании полученных данных и реальной сложности;
- предоставление пользователю рекомендации о сложности следующего маршрута;
- ведение “истории” поездок, включающее в себя:
 - длительность в километрах;
 - время поездки;
 - средняя скорость;
 - максимальная скорость;
 - данные о пульсе (минимальный, максимальный, средний);
 - сложность расчетная;
 - сложность реальная;
- предоставление пользователю статистической информации о прогрессе тренировок.

Дополнительная функциональность:

- возможность моделировать маршруты, для упрощения демонстрации работы приложения.

Требования к пользовательскому интерфейсу:

- возможность вывести пользователю список пройденных маршрутов с визуальным разделением их по оценке.
- графическое отображение статистической информации о прогрессе тренировок.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № «BikeStat: Разработка аналитического приложения с
персональными рекомендациями для велосипедистов»

Протокол получения данных с носимого устройства

Получение измерений с носимого устройства осуществляется запросом GET на адрес https://dt.miet.ru/ppo_it/api/watch.

Для получения данных, во всех запросах к API необходимо указывать заголовок x-access-tokens со значением: az4fvf7nzi1XPIsYiMEu.

Пример корректного ответа от носимого устройства в формате JSON:

```
{  
  "data": {  
    "pulse": {  
      "min": 60, "max": 140, "avg": 110  
    }  
  },  
  "status": "ok"  
}
```

3. Рекомендации к выполнению

- Программный продукт должен работать или на мобильных устройствах, или иметь веб-интерфейс.
- Для осуществления анализа поездок предлагается использовать GeoAPI для построения маршрутов и предоставляемое организаторами API, моделирующее работу носимых устройств, измеряющих пульс велосипедиста для формирования персональных рекомендаций.
- Необходимо предусмотреть сохранение данных о маршрутах в систему управления базами данных (СУБД). Выбор СУБД не регламентируется.
- Следует предусмотреть автоматизацию установки приложения.

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Профиль «Информационные технологии»

Командный кейс № «BikeStat: Разработка аналитического приложения с персональными рекомендациями для велосипедистов»

- В случае реализации в качестве мобильного приложения, должен быть предоставлен установочный файл.
- В случае реализации в качестве WEB-приложения, необходимо предусмотреть автоматическую установку (доставку кода) на целевой машине с подробной инструкцией для системного инженера.
- Разработку рекомендуется вести с помощью системы контроля версий git.
- Рекомендуется использовать unit-тестирование при разработке продукта.

4. Требования к документации

- Титульный лист (с указанием названия кейса и перечислением членов команды);
- Анализ технических требований;
- Обоснование используемых программных средств;
- Структурная и функциональная схемы программного продукта;
- Схема организации данных;
- Описание проведенных испытаний в соответствии с регламентом кейса (снимки экрана и/или запись экрана с работой);
- Программный код (ссылка на репозиторий);
- Подробное описание проекта в репозитории, в том числе инструкция по автоматической установке.
- Видеоролик, демонстрирующий функционирование разработанного устройства в соответствии с регламентом испытаний. На видео необходимо продемонстрировать прохождение каждого, описанного в регламенте, испытаний в соответствии с условиями. При необходимости испытательный полигон подготавливается

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № «VikeStat: Разработка аналитического приложения с
персональными рекомендациями для велосипедистов»**

самостоятельно по предложенной к условию задачи схеме (при его наличии в ТЗ). Видео испытаний готового решения должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя). Видеоролик располагается на стороннем видеохостинге (ВКонтакте, Rutube и др.), ссылка на видеоролик располагается в репозитории в системе контроля версий (может использоваться github, gitlab и др.)

5. Регламент испытаний

1. Построение маршрута.
2. Ведение маршрута, в случае реализации моделирования маршрута, демонстрация различных маршрутов.
3. Демонстрация расчетной оценки маршрута.
4. Демонстрация реальной оценки маршрута на основе данных от носимого устройства.
5. Демонстрация истории пройденных маршрутов.

6. Методические материалы

- <https://www.python.org/>
- <https://nodejs.org/>
- <https://sqlite.org/>
- <https://www.postgresql.org/>
- <https://pypi.org/project/requests/>
- <https://flask.palletsprojects.com/en/2.0.x/>
- <https://reactjs.org/>
- <https://vuejs.org/>
- <https://wiki.openstreetmap.org/wiki/API>
- <https://velofans.ru/raznoe/skorosti-dvizheniya-razlichnyh-velosipedah-raznyh-usloviyah>

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 2 «Торговый бот»

1. Условия

В современном финансовом мире наблюдается рост влияния передовых технологий на рынки и торговлю. Торговый бот - это приложение, позволяющее работать с данными о рынках. Торговые боты становятся все более значимыми инструментами для трейдеров и инвесторов, предоставляя новые возможности для анализа данных и принятия решений.

Подобно другим сферам, где технологии помогают автоматизировать различные процессы, торговые боты в финансовой области играют роль "помощников трейдера". Они спроектированы для анализа данных и принятия автоматизированных решений, позволяя трейдерам адаптироваться к изменчивости рынка.

Каким функционалом могут обладать торговые боты:

1. Анализ трендов: боты могут автоматически анализировать и выявлять текущие тренды на рынке.
2. Принятие решений: боты используют определенные стратегии и алгоритмы для принятия решений о сделках.
3. Мониторинг новостей: боты могут анализировать новостные потоки и отчеты, чтобы выявлять важные события, влияющие на рынок.
4. Управление рисками: боты могут автоматически управлять рисками, ограничивая размеры сделок в соответствии с заданными параметрами.
5. Прогнозирование: боты способны использовать искусственный интеллект или методы технического анализа для предсказания движения рынка.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 2 «Торговый бот»

В рамках этого кейса необходимо обратить внимание на последний пункт. У вас есть возможность создать уникального торгового бота, способного анализировать финансовые данные и прогнозировать движение рынка.

2. Техническое задание

Для реализации задачи необходимо разработать демонстрационное приложение, позволяющее делать выводы о будущем изменении цены на курсы ценных бумаг (криптовалют, акций и т.д.).

Способ реализации приложения остается на усмотрение участников. У экспертов должна быть возможность взаимодействовать с приложением, выбирая определенные параметры для получения прогноза и возможность просмотра результата. От каких параметров будет зависеть результат программы также остается на усмотрение участников. Приложение должно работать на компьютере или на мобильном устройстве.

В приложении обязательно:

- 1) Реализовано хранение данных о текущем положении выбранного рынка ценных бумаг.
- 2) Есть возможность получить результат прогнозирования изменений этого рынка (тренд увеличится, уменьшится или не изменится).

Для реализации прогнозирования можно использовать следующее:

- 1) Предварительная обработка данных;
- 2) Использование обученной вами нейросети или модели машинного обучения для анализа рынка;
- 3) Использование технических методов для анализа рынка.

Результатом работы приложения должно быть прогнозирование изменений значений курсов ценных бумаг (акций, криптовалют и т.д.).

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 2 «Торговый бот»

Приветствуется реализация прогнозирования для нескольких различных рынков.

3. Требования к документации

- Титульный лист (с указанием названия кейса и перечислением членов команды);

- Анализ технических требований;
- Описание используемых методов машинного обучения.
- Описание используемых методов технического анализа.
- Обоснование выбора используемых программных средств;
- Структурная и функциональная схемы программного продукта;
- Описание проведенных испытаний в соответствии с регламентом

кейса (снимки экрана и/или запись экрана с работой);

- Программный код (ссылка на репозиторий).

- Видеоролик, демонстрирующий функционирование разработанного программного продукта в соответствии с регламентом испытаний. На видео или записи экрана необходимо продемонстрировать выполнение каждого, описанного в регламенте, испытания в соответствии с условиями. Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя). Видеоролик располагается на стороннем видеохостинге (ВКонтакте, Rutube и др.), ссылка на видеоролик располагается в репозитории в системе контроля версий (может использоваться github, gitlab, bitbucket и др.)

4. Регламент испытаний

- Запуск приложения;
- Проверка работоспособности программного обеспечения (приложение запускается и работает без технических ошибок);

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ
Профиль «Информационные технологии»
Командный кейс № 2 «Торговый бот»**

- Показ функционала приложения;
- Получение результатов прогнозирования;
- Проверка результатов прогнозирования программного обеспечения;

5. Примерный перечень средств и инструментов для выполнения задания

- <https://tradingview.com>
- <https://www.python.org>
- <https://tensorflow.org>
- <https://pytorch.org>
- <https://pandas.pydata.org/>
- <https://numpy.org/>
- <https://ta-lib.github.io/ta-lib-python/>
- <https://scikit-learn.org/>

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 3 «GridMaster: Разработка интерпретатора
для двумерного перемещения»**

1. Условия

В эпоху цифровизации и роботизации развитие навыков программирования и алгоритмизации становится все более важным. Во многих современных школьных учебных планах появляются курсы по программированию и компьютерным наукам. Один из захватывающих методов погружения учащихся в эту тему — это задача создания интерпретатора для программирования движения исполнителя по двумерной сетке. Разработка интерпретатора для движения исполнителя по двумерной сетке, представляет собой задачу, применимую в областях программирования, робототехники и искусственного интеллекта, где важно умение создавать эффективные алгоритмы и управлять сложными системами. Участникам олимпиады предлагается разработать интерпретатор, способный перемещать исполнителя по полю 21x21 (стартовая координата (0; 0)).

2. Техническое задание

Требуется разработать язык программирования для управления движением исполнителя по двумерной сетке, а также интерпретатор для этого языка. Язык должен поддерживать основные конструкции программирования, включая алгоритмы, циклы, процедуры, условные операторы и переменные. В рамках задачи участникам необходимо создать среду исполнения, которая будет хранить и воспроизводить последовательность команд и координат исполнителя. Среда исполнения должна иметь возможность загрузки команд для исполнителя из .txt файла. Область видимости переменных в этом интерпретаторе глобальная, что означает, что переменные, объявленные в любом месте программы, будут доступны во всех последующих частях программы. Переменные можно объявлять в процедурах, и они сохраняют свои значения на протяжении всей

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 3 «GridMaster: Разработка интерпретатора
для двумерного перемещения»**

программы. Переменные, используемые в блоках IFBLOCK и REPEAT, также имеют глобальную область видимости и будут сохранять свои значения после завершения этих блоков.

Приведем список команд и их описание (N - натуральное число с допустимым диапазоном от 1 до 1,000):

Команда	Описание
RIGHT N	Переместить исполнителя на N клеток вправо.
LEFT N	Переместить исполнителя на N клеток влево.
UP N	Переместить исполнителя на N клеток вверх.
DOWN N	Переместить исполнителя на N клеток вниз.
IFBLOCK DIR	Проверить препятствие в направлении DIR (RIGHT, LEFT, UP, DOWN). Препятствием являются края сетки. Если есть препятствие, выполнить следующие команды до ENDIF.
ENDIF	Завершить блок команд после IFBLOCK DIR.
REPEAT N	Повторить следующую команду (или блок команд до ENDREPEAT) N раз.
ENDREPE AT	Завершить блок команд после REPEAT N.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 3 «GridMaster: Разработка интерпретатора
для двумерного перемещения»**

SET X = N	Задать значение переменной X равным N.
PROCEDURE NAME	Начать определение процедуры* с заданным именем.
ENDPROC	Завершить определение процедуры.
CALL NAME	Вызвать ранее определенную процедуру по имени.

*Процедура - это набор инструкций, который может быть вызван по имени в любом месте программы. Процедуры используются для группировки инструкций в блоки, что делает код более организованным и удобным для понимания и отладки.

А также пример элементарного скрипта на полученном языке:

```
REPEAT 3
  IFBLOCK RIGHT
    RIGHT 5
  ENDIF
ENDREPEAT

DOWN 10

PROCEDURE MoveLeftAndUp
  LEFT 2
  UP 2
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 3 «GridMaster: Разработка интерпретатора
для двумерного перемещения»**

ENDPROC

CALL MoveLeftAndUp

Функциональное задание:

Обязательная функциональность программы:

- Ввод и выполнение команд в соответствии с установленными правилами языка, которые были разработаны для обучения основам программирования учащихся.
- Поддержка основных конструкций языка, включая циклы и условные операторы.
- Возможность объявления и использования переменных.
- Разработка пользовательского интерфейса с окном для редактирования команд, окном для отображения исполнителя на двумерной сетке, а также кнопками "Run" и "Stop" для управления выполнением команд.
- Возможность загрузки исходного кода из .txt файла.
- Поддержка вложенности конструкций языка, таких как вложенные циклы и условные блоки, что позволяет создавать более сложные алгоритмы и улучшает понимание учащимися структур контроля.

Ограничения на вложенность конструкций:

Максимальный уровень вложенности конструкций (например, циклов, условных операторов и процедур) должен быть ограничен до 3 уровней. Это значит, что в любой момент код может иметь не более трех вложенных конструкций.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 3 «GridMaster: Разработка интерпретатора
для двумерного перемещения»**

Пример:

```
REPEAT 5
  IFBLOCK RIGHT
    REPEAT 3
      RIGHT 2
    ENDREPEAT
  ENDIF
ENDREPEAT
```

Дополнительная функциональность:

Создание и вызов процедур:

- Учащиеся должны иметь возможность создавать и вызывать процедуры для группировки и повторного использования кода.

Обработка ошибок и исключений:

- Система должна обрабатывать ошибки и исключения, такие как попытка выхода исполнителя за пределы поля, и предоставлять информативные сообщения об ошибках.

Сохранение и загрузка последовательности команд:

- Предоставить возможность сохранения и загрузки написанных программ в базу данных вместо логирования отдельных команд.

Импорт и экспорт сохраненных программ:

- Учащиеся могут импортировать и экспортировать (из БД в .txt) сохраненные программы для дальнейшего воспроизведения и анализа.

Визуализация движения исполнителя:

- Поддержка графического интерфейса с анимацией для визуального отображения движения исполнителя по сетке.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 3 «GridMaster: Разработка интерпретатора
для двумерного перемещения»**

- В интерфейсе должны присутствовать окно для редактирования команд, окно с отображением движения исполнителя по сетке, а также кнопки "run" и "stop" для управления выполнением программы.

3. Требования к документации

- **Титульный лист:** с указанием названия кейса и перечислением членов команды.

- **Анализ технических требований:** детализация спецификаций и интерпретация задачи. (Интерпретация задачи в данном контексте означает понимание и объяснение целей и требований, предъявляемых к разработке интерпретатора для программирования движения исполнителя по двумерной сетке. Это включает в себя анализ функциональных и дополнительных требований, определение основных и дополнительных возможностей системы, а также понимание желаемого пользовательского взаимодействия и выходных данных, которые должен предоставить разработанный интерпретатор.)

- **Обоснование используемых программных средств:** выбор языков программирования, фреймворков и других инструментов, и их обоснование.

- **Структурная и функциональная схемы программного продукта:** визуализация основных модулей, классов, функций и их взаимодействия.

- **Схема организации данных:** структура данных, диаграммы базы данных или другие релевантные элементы.

- **Описание проведенных испытаний в соответствии с регламентом кейса:** снимки экрана и/или запись экрана с демонстрацией работы программы.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 3 «GridMaster: Разработка интерпретатора
для двумерного перемещения»**

- **Программный код:** ссылка на *Git-репозиторий* с полным набором файлов проекта.
- **README файл:** содержит подробное описание запуска программного обеспечения для разработчика и конечного пользователя.
- **Методы доставки кода:** необходимо предоставить удобные методы для доставки программного обеспечения пользователю. Это может включать в себя готовые установочные пакеты, докер-контейнеры или другие методы, упрощающие процесс установки без необходимости собирать код или устанавливать среду разработки.
- **Подробное описание проекта в репозитории:** объяснение архитектуры, основных модулей и их функциональности, а также любые другие релевантные детали.
- **Видеоролик, демонстрирующий функционирование** разработанного программного продукта в соответствии с регламентом испытаний. На видео или записи экрана необходимо продемонстрировать выполнение каждого, описанного в регламенте, испытания в соответствии с условиями. Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя). Видеоролик располагается на стороннем видеохостинге (ВКонтакте, Rutube и др.), ссылка на видеоролик располагается в репозитории в системе контроля версий (может использоваться github, gitlab, bitbucket и др.).

4. Регламент испытаний

1) Проверка всех функциональных требований, указанных в техническом задании. Каждое требование должно быть проверено на корректность и соответствие спецификациям.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 3 «GridMaster: Разработка интерпретатора
для двумерного перемещения»**

2) Проверка возможности запуска команд из .txt файла. Участники должны демонстрировать, как их разработанный интерпретатор считывает и исполняет команды из внешнего текстового файла.

3) Проверка воспроизведения движения исполнителя по сохраненным координатам. После выполнения команд из текстового файла, система должна позволять сохранять координаты движения исполнителя и затем корректно воспроизводить данный маршрут.

4) Проверка работы интерпретатора на различных задачах. Необходимо тестировать программу на нескольких предоставленных организаторами скриптах из файлов формата .txt для убедительности работы интерпретатора.

5) Проверка обработки исключительных ситуаций.

a) Выход Исполнителя за Границы Сетки:

i) Попытка переместить исполнителя за пределы поля 21x21 (например, команда RIGHT 22 из позиции (0,0)).

b) Неверный Формат Команды:

i) Команды с неправильным синтаксисом (например, RIGH 5 вместо RIGHT 5).

ii) Команды с несуществующими операциями (например, JUMP 5).

c) Некорректные Значения Параметров:

i) Команды с недопустимыми значениями (например, LEFT -5 или UP 1001).

d) Обращение к Неопределенным Переменным:

i) Использование необъявленных переменных в командах (например, SET X = Y, где Y не была объявлена).

e) Ошибки в Циклах и Условных Блоках:

i) Бесконечные циклы (например, REPEAT 0).

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 3 «GridMaster: Разработка интерпретатора
для двумерного перемещения»**

ii) Неправильно оформленные циклы или условные блоки (например, отсутствует ENDREPEAT или ENDIF).

f) Ошибки в Определении и Вызове Процедур:

i) Вызов несуществующих процедур.

ii) Ошибки в определении процедуры (например, повторное объявление процедуры с тем же именем).

g) Превышение Максимального Уровня Вложенности Конструкций:

i) Программы, превышающие ограничение вложенности в 3 уровня.

h) Ошибки Чтения из Файлов:

i) Невозможность считывания команд из поврежденных или отсутствующих .txt файлов.

i) Проблемы Сохранения и Загрузки Программ:

i) Невозможность сохранить или загрузить программу из базы данных.

Для участников кейса по разработке интерпретатора движения исполнителя по двумерной сетке важно уделить особое внимание подготовке видеоматериалов, демонстрирующих корректную обработку каждой из вышеуказанных исключительных ситуаций.

5. Примерный перечень средств и инструментов для выполнения задания

1. [Что такое программирование?](#)
2. [Основы языка Python](#)
3. [Обучение Python для детей и подростков](#)
4. [Основы веб-разработки](#)
5. [Учебник по JavaScript](#)
6. [Понимание топологической сортировки](#)

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 3 «GridMaster: Разработка интерпретатора
для двумерного перемещения»**

7. [Топологическая сортировка в Python](#) (Топологическая сортировка - это процесс сортировки узлов ориентированного графа в линейный порядок, при котором каждый узел предшествует тем узлам, которые он направляет. В контексте нашего кейса, топологическая сортировка может быть использована для определения порядка выполнения процедур и операций, особенно при наличии зависимостей (например, если одна процедура зависит от результатов другой))
8. [Введение в базы данных](#)
9. [Работа с SQLite в Python](#)
10. [Работа с JSON в Python](#)
11. [Git How To](#) - Интерактивное введение в Git.
12. [GitHub Learning Lab](#) - Официальные интерактивные курсы от GitHub.
13. [GitHub Guides](#) - Краткие руководства по основным функциям GitHub.
14. [PLY \(Python Lex-Yacc\)](#) - Инструмент для написания лексических анализаторов и парсеров.
15. [Lark](#) - Современная библиотека парсинга для Python.
16. [autopep8 GitHub](#)
17. [Prettier GitHub](#)
18. [Google Java Format GitHub](#)

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 4 «Сканер уязвимости для NoSql баз данных»

1. Введение

В настоящее время сканеры уязвимостей (или инструменты сканирования защищенности) являются неотъемлемыми средствами, применяемыми специалистами в области информационной безопасности. Рынок предлагает значительное количество таких продуктов, включая как отечественные, так и зарубежные разработки.

Обнаружение уязвимостей в информационных системах, компонентах инфраструктуры или элементах системы информационной безопасности представляет существенную проблему для отделов информационных технологий (ИТ) и информационной безопасности (ИБ). Хотя возможно провести поиск уязвимостей вручную, это является очень трудоемким процессом, который требует значительного времени и существует высокая вероятность пропустить какую-либо уязвимость.

Поэтому наилучшим подходом является использование автоматических инструментов для обнаружения уязвимостей и слабых мест в информационной инфраструктуре предприятия. В настоящее время подобные инструменты обладают широким спектром функциональности. Они позволяют сканировать сети, проводить перебор паролей, обнаруживать неустановленные патчи и уязвимые версии программного обеспечения, находить пароли и учетные записи, установленные по умолчанию, обнаруживать открытые порты и небезопасные службы, а также обнаруживать другие элементы, которые могут создавать угрозы информационной безопасности.

После проведения проверки, большинство сканеров способны создавать отчеты различного формата и назначения. В этих отчетах предоставляется полная картина уязвимостей в инфраструктуре, а также предлагаются рекомендации по их устранению. Каждой уязвимости присваивается

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 4 «Сканер уязвимости для NoSql баз данных»

уникальный номер из базы данных таких уязвимостей, таких как CVE (Common Vulnerabilities and Exposures), NVD (National Vulnerability Database) или Банка данных угроз безопасности информации (БДУ) ФСТЭК России.

Основные функции сканера уязвимостей включают:

1. **Сканирование уязвимостей:** Сканеры выполняют автоматический и систематический поиск уязвимостей в сетях, операционных системах, приложениях и других компонентах информационной инфраструктуры.

2. **Идентификация уязвимостей:** Сканеры анализируют системы и компоненты на предмет известных уязвимостей и проблем безопасности, определяя какие именно уязвимости могут быть использованы злоумышленниками.

3. **Классификация и оценка уязвимостей:** Сканеры классифицируют найденные уязвимости по типу, серьезности и потенциальным последствиям для безопасности. Они также могут предоставлять рекомендации по приоритетам устранения уязвимостей.

4. **Исследование и обнаружение новых уязвимостей:** Некоторые сканеры могут проводить исследования и поиск новых уязвимостей посредством анализа кода, сетевого трафика или других методов, способствующих обнаружению ранее неизвестных уязвимостей.

5. **Генерация отчетов:** Сканеры могут автоматически создавать детальные отчеты о найденных уязвимостях, включающие информацию о типе, серьезности и рекомендации по устранению. Отчеты могут быть представлены в различных форматах для лучшего понимания и использования, как для специалистов по информационной безопасности, так и для руководства.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 4 «Сканер уязвимости для NoSql баз данных»

В последние годы широкое распространение получили нереляционные системы управления базами данных (СУБД), которые используются для хранения и обработки больших объемов данных. В отличие от традиционных SQL-ориентированных баз данных, NOSQL базы данных не используют структурированные таблицы и язык SQL для обработки данных. Вместо этого, они обеспечивают гибкую модель данных, позволяющую хранить слабоструктурированные и неструктурированные данные.

На сегодняшний день с ростом популярности NoSQL баз данных проблема с обнаружением и предотвращением уязвимостей в них становится все более важной задачей. Однако наличие ограниченного числа сканеров уязвимостей для NoSQL баз данных может вызывать трудности при эффективной проверке и обеспечении безопасности таких систем.

1. Условия

Цель: Разработать инструмент сканирования уязвимостей, специализированный для NoSQL баз данных, с целью обнаружения и предотвращения возможных уязвимостей в таких системах.

Основные условия кейса:

- **Определение типа NoSQL баз данных:** Изучить различные типы NoSQL баз данных (документоориентированные, ключ-значение, семейство столбцов, ориентированные на графы) и самостоятельно выбрать, для какого типа будет разрабатываться сканер уязвимостей.
- **Анализ уязвимостей:** Изучить типичные уязвимости, специфичные для выбранного типа NoSQL баз данных. Это может включать такие уязвимости, как недостаточная аутентификация, отказ в обслуживании, обход защиты, инъекции и другие.
- **Разработка сканера уязвимостей:** Создать инструмент, способный автоматически обнаруживать и анализировать типичные уязвимости в

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 4 «Сканер уязвимости для NoSql баз данных»

выбранном типе NoSQL баз данных. Инструмент должен быть специализированным для работы с этим типом баз данных и использовать соответствующие протоколы и API для взаимодействия с базой данных.

- **Процесс сканирования уязвимостей:** Определить процесс сканирования, который включает в себя автоматическую проверку целевой NoSQL базы данных на наличие уязвимостей. Это может включать сканирование конфигурации, анализ структуры данных, проверку правильности доступа, перехват и анализ сетевых пакетов и другие методы сканирования.

- **Генерация отчетов:** Разработать механизм генерации подробных отчетов о обнаруженных уязвимостях. Отчет должен содержать информацию о найденных уязвимостях, их критичности, пути эксплуатации и рекомендации по устранению.

Вам предлагается реализовать веб-приложение, которое будет способно выполнять вышеописанные функции сканера уязвимостей для одного из типа модели данных и структуры хранения NoSQL СУБД.

2. Техническое задание

Для реализации веб-приложения вам необходимо разработать программную часть.

Необходимо предусмотреть в программной части добавление цели сканирования (подразумевается, что пользователь будет вводить IP адрес и порт, на котором запущена СУБД). Информация о цели сканирования должна включать в себя:

- наименование сканирования;
- описание;
- цель сканирования (IP адрес).

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 4 «Сканер уязвимости для NoSql баз данных»

Возможно также добавить дополнительные поля ввода информации для сканирования.

После ввода основных данных пользователь запускает сканирование, которое сохраняется в истории.

После окончания сканирования цели необходимо осуществить демонстрацию полученных данных в виде списка выявленных уязвимостей и графика. График должен представлять цветовую кодировку, которая отражает степень критичности уязвимости. Градация степени риска может быть выбрана самостоятельно. В качестве рекомендации можно взять следующий класс уязвимостей:

- информационная;
- низкая;
- средняя;
- высокая;
- критическая.

При этом у найденных уязвимостей должны быть отображены и реализованы следующие параметры:

- описание;
- рекомендации по устранению уязвимости;
- номер CVE;
- базовая оценка CVE.

Данный перечень не ограничен и при возможности может быть расширен.

После проведения тестирования необходимо предусмотреть возможность выбрать любое сканирование, проведенных ранее, для просмотра уязвимостей в табличном виде и степени критичности уязвимости в виде графика.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 4 «Сканер уязвимости для NoSql баз данных»

Необходимо предусмотреть возможность выгрузки сканирования в формате pdf. По желанию можно расширить список форматов.

3. Обязательные условия при выполнении

- Предусмотреть возможность выявлять уязвимости на основе существующих баз данных уязвимостей CVE, BDU, NVD и других.

- Необходимо предусмотреть модульность программной части, а именно серверную и клиентскую части.

- Предусмотреть сохранение данных, полученных после сканирования в системе управления базами данных (СУБД). Выбор СУБД не регламентируется.

- Разработку программной частей рекомендуется вести с помощью системы контроля версий git.

- Предусмотреть возможность автоматического развертывания приложения с описанием подробной инструкции

4. Рекомендации к выполнению

- Для реализации программной части рекомендуется использовать язык программирования Python, который имеет библиотеки для работы с безопасностью и сканированием уязвимостей.

- Изучить существующие решения сканеров уязвимостей на рынке для лучшего понимания их работы.

- В графическом интерфейсе должны присутствовать следующие разделы и элементы:

- Раздел “Приборная доска”. В данном разделе необходимо отразить 3 последних выполненных сканирования, круговая диаграмма, отражающая соотношение количества выявленных уязвимостей по степени критичности.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 4 «Сканер уязвимости для NoSql баз данных»

○ Раздел “Сканирование”. В нем должны быть отображены поля ввода(Название, описание, Ip-адрес и т.д.). При автоматическом завершении сканирования на экран выводится список выявленных уязвимостей(название, описание, рекомендации по устранению уязвимости, номер CVE, базовая оценка CVE). Также пользователю должна предоставляться возможность скачать отчет по проведенному сканированию (в формате pdf).

○ Раздел “История сканирования”. Содержит список проведенных ранее тестирования. Должна отображаться информация по следующим столбцам: название, описание, количество выявленных уязвимостей, дата проведенного тестирования и т.д. Также должна присутствовать возможность скачать отчет по любому из проведенных сканирований и удаление результатов сканирования.

5. Требования к документации

● Титульный лист (с указанием названия кейса и перечислением членов команды).

● Видеоролик, демонстрирующий функционирование разработанного программного продукта в соответствии с регламентом испытаний. На видео или записи экрана необходимо продемонстрировать выполнение каждого, описанного в регламенте, испытания в соответствии с условиями. Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя). Видеоролик располагается на стороннем видеохостинге (ВКонтакте, Rutube и др.), ссылка на видеоролик располагается в репозитории в системе контроля версий (может использоваться github, gitlab, bitbucket и др.).

● Описание особенностей и аргументация выбранного типа NoSQL СУБД.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 4 «Сканер уязвимости для NoSql баз данных»

- Обоснование выбора языка программирования и используемых программных средств.

- Структурная и функциональная схемы программного продукта.
- Блок-схема работы основного алгоритма.
- Схема базы данных.
- Описание проведенных испытаний в соответствии с регламентом кейса (снимки экрана и/или запись экрана с работой);

- Программный код (ссылка на репозиторий):

- есть краткое описание проекта;
- есть README проекта;
- код проекта соответствует общепринятым стандартам(PEP8).

6. Регламент испытаний

- Производится добавление информации о цели сканирования
- Производится сканирование
- Производится автоматическое завершение сканирования
- Производится просмотр выявленных уязвимостей и графика
- Производится просмотр подробной информации об уязвимости
- Производится просмотр информации уже прошедших сканирований

- Производится выгрузка информации сканирования

7. Примерный перечень средств и инструментов для выполнения задания

- <https://www.python.org/>. Python.org - официальный веб-сайт языка программирования Python. Сайт предоставляет документацию по языку Python, включая руководства, учебники, справочники и официальные PEP (Python Enhancement Proposals).

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 4 «Сканер уязвимости для NoSql баз данных»

- <https://www.djangoproject.com/>. Django.com - официальный веб-сайт фреймворка разработки веб-приложений Django. Сайт Django.com также содержит документацию, которая предлагает пошаговые инструкции по установке Django, а также информацию о его основных компонентах и функциональности
- <https://sqlite.org/>. SQLite.org - официальный веб-сайт базы данных SQLite. Сайт предоставляет подробную документацию, включающую руководства по установке, использованию и настройке SQLite для различных платформ и языков программирования
- <https://www.postgresql.org/>. На сайте вы найдете подробные руководства по установке, настройке, использованию и администрированию PostgreSQL. Документация также включает описание основных SQL команд, функций и возможностей PostgreSQL
- <https://getbootstrap.com/>. Сайт предлагает ресурсы, инструменты, шаблоны и документацию для разработки стильных веб-интерфейсов.
- <https://flask.palletsprojects.com/en/2.0.x/> . Сайт предоставляет ресурсы, документацию и инструменты для разработки веб-приложений с использованием Flask.
- <https://vuejs.org/> . Сайт предоставляет информацию, руководства и ресурсы для разработчиков, интересующихся использованием Vue.js для создания современных веб-приложений. Также предлагаются обучающие материалы, видеоуроки и ресурсы для новичков.
- <https://www.openvas.org/>. OpenVAS - это бесплатная и открытая система сканирования уязвимостей, предназначенная для обнаружения и анализа уязвимостей в информационных системах
- <https://cve.mitre.org/>. Сайт является официальным веб-сайтом CVE (Common Vulnerabilities and Exposures) - общедоступной базы данных

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ
Профиль «Информационные технологии»
Командный кейс № 4 «Сканер уязвимости для NoSql баз данных»**

сведений об уязвимостях. Этот сайт предлагает информацию о различных уязвимостях, которые могут существовать в программном обеспечении, операционных системах, устройствах и других компонентах информационной безопасности.

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Профиль «Информационные технологии»

Командный кейс № 5 «Разработка распределенного хранилища данных»

1. Условия

Системы хранения данных активно развивались все последние годы и прошли путь от индивидуальных файлов до больших распределенных хранилищ данных. Распределенная структура позволяет снизить риски утери данных и ускорить их обработку. В рамках данного кейса обучающимся предлагается разработать распределенное хранилище данных, позволяющее выполнять простые аналитические запросы.

Разработать требуется также и интерфейс администратора, с помощью которого можно будет осуществлять мониторинг различных показателей системы.

2. Техническое задание

Требуется разработать распределенное хранилище данных, которое будет предоставлять пользователю возможность хранить данные в виде текстовых файлов и осуществлять полнотекстовый поиск по этим файлам. Система должна поддерживать функции загрузки и удаления файлов, а также функцию поиска. Хранилище состоит из центрального сервера и серверов-хранилищ. Сервера-хранилища дублируют данные, они хранят копии одних и тех-же файлов. Центральный сервер содержит базу данных с информацией о расположении файлов в системе и их версиях. Пользователь взаимодействует с системой через центральный сервер, именно он управляет действиями серверов-хранилищ. Помимо этого, должна быть поддержана роль администратора, который может видеть текущее состояние базы данных и при необходимости вносить соответствующие изменения.

Функциональное задание:

1.1. Система представляет собой веб-интерфейс пользователя, используя который можно:

1.1.1. Авторизоваться

1.1.2. Добавить файл

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Профиль «Информационные технологии»

Командный кейс № 5 «Разработка распределенного хранилища данных»

1.1.3. Удалить файл

1.1.4. Изменить файл (обновить старый файл на новый)

1.1.5. Выполнить простой запрос: поиск подстроки в строках файла.

1.1.6. Выйти из системы

1.2. Дополнительно, система предоставляет функционал администратора в виде веб-страницы, используя который можно:

1.2.1. Авторизоваться

1.2.2. Добавить/удалить/изменить информацию о пользователе

1.2.3. Получить список пользователей с указанием, когда они в последний раз заходили в систему и какие операции они в системе выполняли

1.2.4. Получить список файлов с указанием пользователей-владельцев (с указанием, на каких серверах системы файлы каких версий хранятся)

1.2.5. Подключить/отключить сервер-хранилище

1.2.6. Выйти из системы

1.3. Система состоит из центрального сервера, который обеспечивает:

1.3.1. Работу веб-интерфейсов пользователей и администраторов

1.3.2. Распределение задач на чтение/обновление/удаление файлов с дополнительных серверов-хранилищ

1.3.3. Распределение задач на получение данных (центральный сервер распределяет задачи по получению данных поровну между серверами-хранилищами)

1.4. Система состоит из неограниченного количества дополнительных серверов-хранилищ, которые обеспечивают:

1.4.1. Хранение файлов от пользователей (с контролем их версий)

1.4.2. Выполнение запросов пользователей по указанной части файла (часть указывается центральным сервером)

1.4.3. Для простоты считаем, что сервера-хранилища достаточно надежные и не обеспечиваем им дополнительные алгоритмы обнаружения проблем с

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Профиль «Информационные технологии»

Командный кейс № 5 «Разработка распределенного хранилища данных»

ними. Сервера хранилища настраиваются только через интерфейс пользователя или конфигурационные файлы (см. критерии оценивания)

2. Разъяснения логики работы хранилища.

2.1. **Файлы, которые хранит хранилище.** ASCII-текстовые многострочные файлы. Размер файла может быть очень большим (ограничением служит ограничения файловой системы).

2.2. **Помещение файла внутрь хранилища (обновление файла).** Пользователь, используя центральный сервер выбирает файл и диска и помещает его в хранилище. Центральный сервер используя TCP-сеть последовательно пересылает файл на все сервера-хранилища. Если такой файл уже есть на серверах-хранилищах - он заменяется на новый. Пока идет пересылка, центральный сервер может принимать запросы от пользователей на пересылку еще нескольких файлов или запросы на поиск в хранилище.

2.3. **База данных.** Центральный сервер сохраняет имя файла, его идентификатор, список серверов-хранилищ с актуальными идентификаторами файлов на них, а также, размер файла.

2.4. **Обновление версий файлов внутри хранилища.** Центральный сервер и Сервера-хранилища поддерживают простую версиюность файла в виде идентификатора. Первоначально, помещая файл на сервера-хранилища центральный сервер устанавливает идентификатор файла в единицу. Каждый раз, когда файл обновляется пользователем, идентификатор увеличивается на единицу.

2.5. **Список файлов, хранящихся внутри хранилища.** Интерфейс центрального сервера должен позволять отображать все файлы, которые хранятся в хранилище.

2.6. **Удаление файлов.** Пользователь, используя центральный сервер может отправить запрос на удаление файла. При этом, центральный сервер,

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Профиль «Информационные технологии»

Командный кейс № 5 «Разработка распределенного хранилища данных»

используя TCP-сеть последовательно удаляет файлы со всех серверов-хранилищ.

2.7. Поиск в хранилище. Пользователь, используя интерфейс центрального сервера может отправить запрос на поиск подстроки внутри строк текста. Для простоты считаем, что поиск может вестись только внутри одной строки. Центральный сервер, зная текущую версию файла и его размер, и зная, на каких серверах-хранилищах эта версия уже находится, условно делит весь файл на равные части и отправляет задания на поиск каждому серверу-хранилищу. Каждый сервер-хранилище осуществляет поиск внутри указанных сервером частей файла. Пример: файл имеет размер 100 байт. Хранится он на двух серверах-хранилищах. Центральный сервер отправляет запрос на поиск первому серверу хранилищу с указанием на поиск с 1 по 50 байт, второму серверу с 50 до 100 байта. Так как у нас идет поиск по строкам, то первый сервер фактически ищет подстроку с 1 по 50 байт + до окончания строки после 50 байта. Второй сервер-хранилище ищет подстроку, начиная с первого символа первой строки после 50 байта. Оба сервера хранилища возвращают строки, в которых найдена искомая подстрока.

3. Требования к документации

- Титульный лист (с указанием названия кейса и членов команды - фио и класс)
- Описание логики работы системы
- Структурная и функциональная схемы системы
- Схема организации данных
- Описание проведенных испытаний в соответствии с регламентом кейса (снимки экрана и/или запись экрана с работой)
- Блок-схемы работы основных алгоритмов
- Программный код (ссылка на репозиторий)
- Подробное описание проекта в репозитории

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Профиль «Информационные технологии»

Командный кейс № 5 «Разработка распределенного хранилища данных»

- Видеоролик, демонстрирующий функционирование разработанного программного продукта в соответствии с регламентом испытаний. На видео или записи экрана необходимо продемонстрировать выполнение каждого, описанного в регламенте, испытания в соответствии с условиями. Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя). Видеоролик располагается на стороннем видеохостинге (ВКонтакте, Rutube и др.), ссылка на видеоролик располагается в репозитории в системе контроля версий (может использоваться github, gitlab, bitbucket и др.).

4. Регламент испытаний

- Демонстрация функций добавления/обновления/удаления файлов с авторизацией
- Демонстрация выполнения запросов на поиск данных с указанием времени выполнения запроса в случае присутствия одного сервера-хранилища и нескольких серверов-хранилищ. Разница во времени выполнения должна быть видна (то есть необходимо для примера взять такие данные, чтобы разница в скорости выполнения запроса при одном сервере и при двух различалась кратно или на порядок).
- Демонстрация работы интерфейса администратора

5. Примерный перечень средств и инструментов для выполнения задания

- Что такое распределенные хранилища (<https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%B>

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**

Профиль «Информационные технологии»

Командный кейс № 5 «Разработка распределенного хранилища данных»

[D%D0%BD%D0%B0%D1%8F %D0%B1%D0%B0%D0%B7%D0%B0 %D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85\)](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%B%D0%BD%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F)

- Что такое распределенные вычисления (<https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%B%D0%BD%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F>)
- Как работают распределенные системы (<https://www.block-chain24.com/articles/что-такое-распределенные-системы-и-как-они-работают>)
- Основные понятия распределенных вычислений (<https://studfile.net/preview/16452933/page:21/>)
- Введение в сетевое программирование на Питон (<https://python-course.readthedocs.io/projects/year2/en/latest/lessons/07-sockets.html#:~:text=%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F,%2D%20%D1%81%D0%BE%D0%BA%D0%B5%D1%82%2C%20%D0%BA%D0%BE%D1%82%D0%BE%D1%80%D1%8B%D0%B9%20%D0%BF%D1%80%D0%B8%D0%BD%D0%B8%D0%BC%D0%B0%D0%B5%D1%82%20%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D1%8F.>)
- Начинающему сетевому программисту C++ (<https://habr.com/ru/articles/582370/>)
- Программирование WebSockets (<https://otus.ru/nest/post/678/>)
- Программирование сокетов C# (https://professorweb.ru/my/csharp/web/level3/3_1.php)

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 6 «Sea Battle: Разработка Web-игры по принципу
морского боя для розыгрыша призов»**

1. Условия

В современном мире маркетинг играет ключевую роль в успешной коммерческой деятельности, а эффективные маркетинговые стратегии обеспечивают компаниям конкурентное преимущество, помогая установить сильный бренд, налаживая коммуникацию с клиентами и адаптируясь к постоянно меняющимся требованиям рынка. Одной из таких стратегий является геймификация. На примере разработки web-игры по принципу морского боя школьникам предстоит погрузиться в маркетинг геймификации.

Суть геймификации, как маркетингового инструмента, web-игры морской бой заключается, например, в стимулировании пользователей купить большее количество товаров в интернет-магазине компании. За каждую покупку на определенную сумму, пользователю будут начисляться бонусные выстрелы. Эти выстрелы пользователь может тратить на поле морского боя и, если он попадет в корабль, то получит какой-либо гарантированный приз.

С одной стороны системы выступает администратор, который создает игровые поля, расставляет на ней корабли и назначает за каждым кораблем свой уникальный приз, а также начисляет игрокам (клиентам) выстрелы. С другой стороны, клиент компании, который получает доступ к игровому полю и может совершать выстрелы по кораблям в попытках заполучить заветный приз.

Для реализации данной web-игры предлагается использовать различные системы разграничения прав доступа, работу с пользовательскими сессиями, а также систему организации очередей запросов.

2. Техническое задание

Требуется разработать web-приложение в формате игры «морской бой» с разграничением уровня доступа к функционалу системы в зависимости от

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 6 «Sea Battle: Разработка Web-игры по принципу
морского боя для розыгрыша призов»**

выбранной роли пользователя. Система должна поддерживать 2 роли: администратор и пользователь (игрок в морской бой).

Функциональное задание:

Обязательная функциональность программы со стороны администратора:

- Регистрация и авторизация администратора в системе
- Создание игрового поля в формате «морской бой» - квадрат размерностью $N \times N$ и с описанием правил игры
 - Где N – параметр, который может указывать администратор
 - Описание правил игры должно поддерживать любой язык разметки для более структурированного представления информации
- Просмотр, редактирование и удаление игровых полей
- Создание, редактирование, просмотр и удаление списка призов
 - Предусмотреть название, описание и изображение приза
 - Предусмотреть возможность фильтрации призов по различным параметрам, в том числе по тем призам, которые выиграла пользователи
- Создание, удаление и просмотр кораблей
 - Предусмотреть создание ТОЛЬКО однопалубных кораблей (1x1)
 - За каждым кораблем ОБЯЗАТЕЛЬНО должен быть закреплен приз, без этого корабль нельзя создать и разместить на поле
- Расстановка кораблей на игровое поле
 - Нельзя ставить корабли в одну и ту же ячейку
- Предусмотреть добавление и удаление пользователя на игровое поле по его уникальному идентификатору
 - Каждый пользователь должен иметь свой уникальный ID
- Предусмотреть начисление выстрелов для пользователя на конкретном игровом поле

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 6 «Sea Battle: Разработка Web-игры по принципу
морского боя для розыгрыша призов»**

○ Количество выстрелов у одного и того же пользователя на различных игровых полях может отличаться

● Важно, администраторы могут только начислять выстрелы пользователям. Стрелять по игровому полю администраторы не могут

Обязательная функциональность программы со стороны пользователя:

● Регистрация и авторизация пользователя в системе

● Просмотр и возможность зайти в заранее созданное администратором игровое поле

● Возможность осуществить выстрел на игровом поле

○ Если у пользователя нет выстрелов – уведомить его об этом и запретить совершать выстрел

○ В случае промаха (не попал в корабль) или успешном попадании (попал в корабль) уведомить об этом различными сообщениями

● Просмотр выигранных пользователем призов на отдельной странице

Дополнительная функциональность:

● Предусмотреть запрет редактирования и удаления игрового поля после совершения хотя бы одного выстрела

● Предусмотреть проверку на запрет выстрела в одинаковую клетку игрового поля

● Предусмотреть запрет удаления пользователя с игрового поля после совершения им хотя бы одного выстрела

Краткое описание работы системы – Администратор регистрируется в системе, после чего авторизуется и создает игровое поле размером 10 на 10 клеток. После чего создает два приза, например, подарочные сертификаты, номиналом 1000 и 2000 рублей соответственно. После чего создает два корабля и закрепляет за ними призы. Далее администратор выставляет

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 6 «Sea Battle: Разработка Web-игры по принципу
морского боя для розыгрыша призов»**

корабли на игровое поле и приглашает 2-х пользователей по заранее известным уникальным идентификаторам. Кроме этого, администратор начисляет пользователям по 2 выстрела на игровом поле. Пользователи, которые по аналогии с администратором, заранее создали аккаунты и вошли в систему, получают доступ к новому игровому полю. Зайдя на него они получают возможность выстрелить по 2 раза каждый.

3. Требования к документации

- Титульный лист (с указанием названия кейса и перечислением членов команды);
- Анализ технических требований;
- Обоснование используемых программных средств;
- Структурная и функциональная схемы программного продукта;
- Схема организации данных;
- Описание проведенных испытаний в соответствии с регламентом кейса (снимки экрана и/или запись экрана с работой);
- Программный код (ссылка на репозиторий);
- Видеоролик, демонстрирующий функционирование разработанного устройства в соответствии с регламентом испытаний. На видео необходимо продемонстрировать прохождение каждого, описанного в регламенте, испытаний в соответствии с условиями. При необходимости испытательный полигон подготавливается самостоятельно по предложенной к условию задачи схеме (при его наличии в ТЗ). Видео испытаний готового решения должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя). Видеоролик располагается на стороннем видеохостинге (ВКонтакте, Rutube и др.), ссылка на видеоролик

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 6 «Sea Battle: Разработка Web-игры по принципу
морского боя для розыгрыша призов»**

располагается в репозитории в системе контроля версий (может использоваться github, gitlab и др.).

- Схемы алгоритмов ключевых модулей системы:
 - Разграничение прав доступа (пользователь, админ)
 - Механика создания игрового поля
 - Механика расстановки кораблей и закрепления за ними призов
 - Механика осуществления выстрела пользователем (с проверкой возможности выстрела)

- Подробное описание проекта в репозитории:
 - Метод установки и/или запуска программного продукта
 - Описание структуры данных
 - Описание функциональных блоков программного продукта
 - Скриншоты интерфейсов программного продукта

4. Регламент испытаний

Производится запуск программного продукт в виде web-приложения

Производится регистрация пользователя в системе

Производится регистрация и авторизация администратора в системе

Администратор создает игровое поле размерность $N \times N$, N - указывает проверяющий

Администратор создает несколько призов. Количество призов определяется проверяющим

Администратор создает несколько кораблей. Количество кораблей определяется проверяющим

Администратор расставляет корабли на игровое поле

Администратор добавляет созданного в пункте в пользователя на игровое поле

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 6 «Sea Battle: Разработка Web-игры по принципу
морского боя для розыгрыша призов»**

Администратор начисляет пользователю несколько выстрелов
(минимум 2)

Производится авторизация пользователя в системе

Пользователь заходит на поле и осуществляет один выстрел в корабль,
один выстрел мимо и один выстрел, когда выстрелы закончились

Пользователь смотрит список своих призов

Демонстрируется работа unit-тестов.

**5. Примерный перечень средств и инструментов для
выполнения задания**

1. Что такое геймификация? <https://gb.ru/blog/gejmifikatsiya/>
2. Алгоритмы и структуры данных -
<https://stepik.org/course/156/promo>
3. Базы данных и для чего их используют? -
<https://skillbox.ru/media/code/kakie-bazy-dannykh-sushchestvuyut-i-dlya-chego-ikh-ispolzuyut-sql-nosql-olap/>
4. Основы работы с MySQL - <https://www.webmasterwiki.ru/MySQL>
5. Основы языка python - <https://pythonworld.ru/samouchitel-python>
6. Основы web-разработки -
https://professorweb.ru/my/html/html5/level1/web_info.php

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 7 «Разработка десктопного приложения для анализа и
прогнозирования климатических данных»**

1. Условия

В современном мире проблемы, связанные с изменением климата и погодными явлениями, становятся все более актуальными и остро обсуждаемыми. С глобальным потеплением, катастрофическими стихийными бедствиями и изменением экосистем, существует необходимость в разработке новых подходов для анализа и прогнозирования климатических данных и погоды. Наука и технологии машинного обучения предоставляют нам мощные инструменты для эффективного изучения и понимания сложных климатических систем.

Целью данного кейса является разработка десктопного приложения, предназначенного для интерактивного анализа и прогнозирования климатических данных и погодных явлений. При разработке приложения требуется использовать методы обработки данных и статистического анализа, а также машинное обучение для создания более точных и надежных прогнозов погоды и климата.

Участникам Олимпиады предлагается разработать десктопное приложения для анализа и прогнозирования климатических данных, погодных явлений. Разрабатываемое приложение должно обладать рядом ключевых функций, которые позволят производить анализ и прогнозировать климатические данные и погодные явления. Для пользователя должна быть реализована возможность визуализировать данные в виде графиков и диаграмм, проводить анализ климатических данных для выявления трендов, а также получать прогнозы погоды на ближайшие дни или недели. Для обеспечения сохранения данных и персонализированного доступа предусмотрена система авторизации пользователей. Кроме того, приложение позволяет экспортировать данные и результаты анализа для дальнейшего использования в других приложениях или научных исследованиях.

Работа приложения должна производиться как минимум с температурой, влажностью, осадками и ветром.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 7 «Разработка десктопного приложения для анализа и
прогнозирования климатических данных»**

2. Техническое задание

Требуется разработать приложение для анализа и прогнозирования климатических данных. С интуитивно понятным графическим интерфейсом. Платформа для разработки должна обязательно включать в себя минимум одну из перечисленных:

- Windows
- MacOS
- Linux

Климатические данные хранятся по ссылке:

<https://drive.google.com/drive/folders/1Eh4j3MFveUR7yi3q3Ts4VWfggw3t3hMX?usp=sharing>

В папке находятся два файла (каталог и описание – помогут в навигации по данным) и папка с даннфми в форматах .dat.

Технические требования:

1. Загрузка и хранение данных:
 - возможность загрузки и хранения разнообразных климатических данных, включая температуру, осадки, влажность, скорость ветра и др (максимальный объем данных не должен превышать 10 гб);
 - поддержка различных форматов данных для импорта и экспорта (CSV и XLSX);
2. Визуализация данных:
 - отображение климатических данных на графиках, диаграммах (с возможностью задать временной отрезок и место) ;
 - возможность выбора временного интервала и места (города или метеостанции) для анализа;
3. Анализ и обработка данных:

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 7 «Разработка десктопного приложения для анализа и
прогнозирования климатических данных»**

- выявление трендов(долгосрочные изменения в погодных условиях), и корреляций между различными погодными параметрами;
 - проведение статистического анализа данных для выявления закономерностей (нахождение средних значений, медиан, отклонений);
4. Прогнозирование погоды:
- реализация моделей машинного обучения для прогнозирования погоды на ближайшие дни или недели (точность не оценивается, метод прогнозирования на усмотрение участников);
5. Мониторинг погодных явлений:
- возможность подключения к внешним источникам данных о погоде и климате (рекомендуется использовать следующий API:
<https://www.gismeteo.ru/api/>);
 - вывод текущей информации о климатической ситуации в отдельной вкладке (окне);
 - данные для вывода:
 - день, время, локация;
 - температура, влажность;
 - ветер;
 - давление;
 - облачность;
6. Настройки и персонализация:
- возможность настройки параметров анализа и прогноза в соответствии с потребностями пользователя (например: изменение промежутка времени для анализа и прогнозирования);
7. Система уведомлений:

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 7 «Разработка десктопного приложения для анализа и
прогнозирования климатических данных»**

- вывод Push-уведомлений: для предупреждения пользователя на основе обнаруженных погодных изменений (вывод на экран или в самом приложении уведомлений);

8. Безопасность и доступ:

- авторизация пользователей через БД (хранение логина и пароля в БД)

;

9. Эмулятор данных для тестирования (дополнительный пункт):

- возможность генерации данных для проверки функциональности и производительности

- данные для генерации:

- температура

- влажность

- давление

Требования к графическому интерфейсу:

- главное меню: должно содержать основные разделы приложения, такие как “Работа с файлами”, “Анализ”, “Визуализация”, “Прогнозирование”, “Мониторинг”, “Выход”

- панель навигации для быстрого доступа между разделами

- окно загрузки данных:

- должно предоставлять возможность загрузки климатических данных с локального компьютера или через ссылку.

- пользователь должен иметь возможность выбирать формат данных (CSV, XLSX) ;

- раздел визуализации данных:

- позволяет пользователю выбирать типы данных для отображения (графики, диаграммы) и временной отрезок.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 7 «Разработка десктопного приложения для анализа и
прогнозирования климатических данных»**

- обеспечивает выбор местоположения для анализа (города или метеостанции).
 - раздел анализа данных:
 - должны быть представлены в удобном для восприятия виде, таком как графики, диаграммы, и текстовые выводы.
 - пользователь должен иметь возможность сохранять результаты анализа.
 - раздел прогнозирования:
 - позволяет выбрать временной интервал для прогноза (дни или недели).
 - отображает прогнозные данные и сравнивает их с реальными значениями.
 - раздел мониторинга погодных явлений:
 - включает информацию о текущей климатической ситуации из внешних источников.
 - должно предоставлять вкладку или окно для мониторинга текущей погоды.
 - масштабируемость:
 - графический интерфейс должен быть масштабируемым;
- 3. Требования к документации**
 - титульный лист (с указанием названия кейса и перечислением членов команды);
 - анализ технических требований;
 - структурная и функциональная схемы программного продукта;
 - блок-схема работы основного алгоритма;
 - описание проведённых испытаний в соответствии с регламентом кейса (снимки экрана и/или запись экрана с работой);

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 7 «Разработка десктопного приложения для анализа и
прогнозирования климатических данных»**

- программный код (ссылка на репозиторий).
- видеоролик, демонстрирующий функционирование разработанного устройства в соответствии с регламентом испытаний. На видео необходимо продемонстрировать прохождение каждого, описанного в регламенте, испытаний в соответствии с условиями. При необходимости испытательный полигон подготавливается самостоятельно по предложенной к условию задачи схеме (при его наличии в ТЗ). Видео испытаний готового решения должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя). Видеоролик располагается на стороннем видеохостинге (ВКонтакте, Rutube и др.), ссылка на видеоролик располагается в репозитории в системе контроля версий (может использоваться github, gitlab и др.)

4. Регламент испытаний

1. Запуск установочного файла (требуется исполняемый файл);
2. Проверка работоспособности приложения:
 - a. производится анализ и визуализация климатических данных;
 - b. присутствует функциональность прогнозирования, присутствуют метрики;
 - c. приложение работает без вылетов;
3. Демонстрация возможностей приложения
4. Верификация корректной визуализации данных (присутствует возможность выбрать вид визуализации (графики или диаграммы), а также выбрать временной промежуток, за который происходит визуализация данных);
5. Тестирование прогнозирования ;
6. Тестирование получения климатических данных в реальном времени;
7. Тестирование персонализации и экспорта данных;

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
**Командный кейс № 7 «Разработка десктопного приложения для анализа и
прогнозирования климатических данных»**

5. Примерный перечень средств и инструментов для выполнения задания

- **Язык программирования:** Python.
- **Интерфейс:** PyQt (<https://riverbankcomputing.com/software/pyqt/intro>) или Tkinter (<https://docs.python.org/3/library/tkinter.html>) для создания интерфейса.
- **Анализ данных:** Библиотеки Pandas (<https://pandas.pydata.org/>) и NumPy (<https://numpy.org/>).
- **Визуализация:** Matplotlib (<https://matplotlib.org/>) или Seaborn (<https://seaborn.pydata.org/>) для создания графиков и визуализации данных.
- **Погодные данные:** Публичные API, например, OpenWeatherMap (<https://openweathermap.org/api>).
- **Базы данных:** SQLite (<https://www.sqlite.org/index.html>) или MySQL (<https://www.mysql.com/>).
- **API для мониторинга:** <https://www.gismeteo.ru/api/>

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 8«Умный домофон»

1. Условие

В современном мире, где технологии становятся все более важной частью нашей жизни, умные устройства уверенно набирают популярность. Одним из ключевых элементов умного дома является умный домофон, который позволяет владельцам контролировать доступ дистанционно — видеть посетителей, а также открывать им дверь (шлагбаум, калитку) прямо со смартфона.

В рамках данного кейса вам предстоит разработать нативное мобильное приложение для умного домофона. Приложение должно быть разработано для операционной системы Android и должно предоставлять возможность пользователю взаимодействовать с домофоном в реальном времени, получать уведомления о вызовах с домофона, просматривать историю использования домофона, открывать дверь дистанционно и настраивать параметры домофона.

2. Техническое задание

Необходимо разработать нативное Android-приложение в Android Studio / IntelliJ IDEA, которое будет обеспечивать:

1. Просмотр статуса домофона в реальном времени.
2. Получение уведомлений о входящих вызовах через домофон.
3. Просмотр истории входящих вызовов.
4. Открытие двери дистанционно.
5. Задание параметров приложения и домофона.

Рассмотрим подробнее экраны приложения.

Главный экран

Данный экран необходимо показывать при запуске приложения. Экран может находиться в одном из следующих состояний:

- **Первый вход.** Данное состояние экрана выводится в случае, если параметры домофона были не настроены. В этом состоянии на экране необходимо отобразить следующие элементы:

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**

**Профиль «Информационные технологии»
Командный кейс № 8«Умный домофон»**

- Приветственный текст, где пользователю объясняется, что необходимо сделать.
- Кнопка с уникальным идентификатором `button_start`. По нажатию на эту кнопку необходимо перейти к экрану настроек домофона.
- **Ошибка подключения.** Данное состояние отображается при некорректных настройках домофона или любой другой сетевой ошибке (нет подключения к Интернету). В этом состоянии необходимо отобразить следующие элементы:
 - Информацию о том, что произошла ошибка.
 - Кнопку с уникальным идентификатором `button_retry`. По нажатию на эту кнопку необходимо совершить повторную попытку подключения, предварительно скрыв данное состояние и заменив на загрузку.
 - **Информация о домофоне.** Данное состояние отображается при успешном подключении к домофону и содержит следующие элементы:
 - Текстовое поле с идентификатором `text_intercom_model` в котором отображена информация о подключённой модели домофона.
 - Кнопка с уникальным идентификатором `button_retry`, которая обновляет полученную информацию, предварительно скрывая данное состояние, заменяя его загрузкой.
 - Элемент вывода изображения с подсостояниями:
 - Получение первого фото, содержащее кнопку с уникальным идентификатором `button_take_photo`, для запроса фотографии с домофона. При нажатии состояние элемента (а не всего экрана) заменяется на загрузку.

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Профиль «Информационные технологии» Командный кейс № 8 «Умный домофон»

- Отображение фото. Состояние, на которое выводится изображение из домофона на элемент с идентификатором `image_intercom`. Помимо этого, состояние содержит кнопку с уникальным идентификатором `button_take_photo`, для повторного запроса фотографии с домофона. При нажатии состояние элемента (а не всего экрана) заменяется на загрузку.

- В случае ошибки необходимо вывести состояние “Получение первого фото” с дополнительным текстовым полем, с идентификатором `text_error`, поясняющим о том, что произошла ошибка.

- **Состояние загрузки** - это состояние с выводом для пользователя любой информации, означающей о необходимости ожидания получения информации.

Также, независимо от состояния экрана, на нём необходимо отобразить следующие элементы:

- Кнопку с идентификатором `button_settings`, открывающую экран настроек домофона.

- Кнопку с идентификатором `button_history`, открывающую экран истории вызовов.

Экран настроек

Данный экран содержит информацию о введённых настройках (если есть) приложения с возможностью изменения. Для ввода настроек существуют следующие поля ввода:

- Поле ввода номера дома с идентификатором `input_house`. Ограничения для ввода: не более 4-х символов (включительно). Можно ввести только:

- цифры от 0 до 9 (только в начале);
- символ “/” (не более одной, не может быть в начале и в конце);

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 8«Умный домофон»

- английские буквы от а до е включительно (не более одной, не может быть в начале).

- Поле ввода квартиры с идентификатором `input_flat`. Ограничения для ввода, не более 6-и символов (включительно). Можно ввести только цифры.

Для сохранения настроек присутствует кнопка с идентификатором `button_save`, при нажатии на которую производится сохранение введённых параметров. Валидация полей ввода должна происходить мгновенно. Если данные некорректны или соответствуют уже существующим - то кнопка должна быть деактивирована (но отображаться).

Возврат к главному экрану должен происходить через навигационную (аппаратную) кнопку назад, а также любую дополнительную кнопку на экране (например, в тулбаре). При обновлении настроек и возврате в главному экрану - главный экран должен автоматически обновляться.

Экран истории вызовов

Данный экран содержит список последних вызовов. Список должен быть реализован с помощью компонента *RecyclerView* с идентификатором `recycler`. Каждый элемент внутри должен содержать следующие компоненты:

- Текстовое поле с идентификатором `text_date`, который содержит время и дату в формате ЧЧ:ММ ДД.ММ.ГГГГ (например, если звонок был в 9 часов утра 5го июня 2023 года, то необходимо вывести 09:00 05.06.2023).

- Текстовое поле с идентификатором `text_status`, который содержит один из статусов результата вызова: *открыт, сброшен*.

Вся история вызовов хранится локально на смартфоне и **не** передаётся на сервер. Разрешается дополнить данный список дополнительными элементами, например, метками времени.

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Профиль «Информационные технологии» Командный кейс № 8 «Умный домофон»

Возврат к главному экрану должен происходить через навигационную (аппаратную) кнопку назад, а также любую дополнительную кнопку на экране (например, в тулбаре).

Экран вызова

Данный экран появляется при вызове на домофон и содержит следующие элементы:

- Кнопку с идентификатором `button_open`, открывающую дверь.
- Кнопку с идентификатором `button_close`, сбрасывающую вызов.

Уведомления

При пропущенном вызове необходимо вывести уведомление, в котором будет информация выведено текстовое сообщение в формате: *Пропущенный вызов*.

Общие требования к мобильному приложению

Все элементы должны иметь поддержку доступности для экранных дикторов, а также иметь достаточные контрастность и размер.

API сервера и серверное взаимодействие

В данном кейсе Вам будет предоставлен готовый сервер, к которому Вы можете подключаться и тестировать приложение. Сервер осуществляет работу с помощью RESTful API, в котором существуют следующие запросы:

Тип запроса	Путь	Ответ (JSON)
GET	/info	Статус 200: <i>model: String</i> Статус 400 (введены неверные данные) Статус 403 (не

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ

Профиль «Информационные технологии»
Командный кейс № 8 «Умный домофон»

		хватает параметров авторизации)
GET	/image	Статус 200: <фото бинарными данными> Статус 400 (введены неверные данные) Статус 403 (не хватает параметров авторизации)
POST	/call	В теле необходимо указать: <i>status</i> с одним из следующих значений: - <i>open</i> - для открытия - <i>close</i> - для закрытия Ответ: Статус 200 Статус 400 (введены неверные данные) Статус 403 (не хватает параметров

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**

**Профиль «Информационные технологии»
Командный кейс № 8 «Умный домофон»**

		авторизации)
POST (для разработчиков)	/make_call	Создать входящий звонок. Использовать только в Postman или аналогичных системах.

Ко всем запросам необходимо в *header* добавить параметры:

Ключ: *house*. Значение: <номер дома>

Ключ: *flat*. Значение: <номер квартиры>

Помимо данных запросов для связи с домофоном присутствует соединение с помощью SOCKET. В шаблоне заготовке Вам будет предоставлена реализация данного соединения для получения вызова.

Рекомендуемые подходы:

- Архитектура MVP/MVI/MVVM, отсутствие лишней логики в слое UI;
- Retrofit/Volley/Cronet для работы с API домофона;
- Локальное кэширование данных при помощи SQLite, Room, Realm (на выбор);
- Объявление строковых ресурсов, ресурсов типа color, dimen и т.п., если эти значения используются многократно.

Дизайн приложения:

- Рекомендуется использовать Material Design 3;
- Рекомендуется нарисовать собственную иконку приложения.

Приложение должно корректно работать на устройстве с ОС Android 10.0 и выше. Все строковые ресурсы должны быть локализованы только для русского языка. Максимальная задержка в обработке событий приложением не должна превышать 10 секунд. Ограничение по памяти для файла apk: 100 Мб.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 8«Умный домофон»

Все действия должны происходить без подтверждающих диалоговых окон/экранов.

Базовое (baseline) решение представлено в виде проекта Android Studio / IntelliJ IDEA: innovationcampus.ru/lms в курсе «Московская предпрофессиональная олимпиада». В нем продемонстрирована навигация по приложению. Используйте этот проект, не изменяя пакет приложения и файлы сборки (build.gradle.kts).

Ваше решение будет валидироваться в системе автоматического тестирования Test Lab.

Сдать в Test Lab необходимо zip архив, в корневом каталоге которого располагается дополненный проект Android Studio/IntelliJ IDEA. Максимальный размер zip-архива: 150 Мб. После загрузки zip архива приложение будет запущено в Test Lab для проверки на соответствие техническому заданию олимпиады. Test Lab валидирует работу приложения почти одновременно на 3–7 устройствах.

При сборке проекта все изменения в файле build.gradle.kts будут проигнорированы – будут использованы зависимости только из базового решения (в больших индустриальных проектах подключению новых библиотек предшествует детальный анализ со стороны инженеров).

3. Требования к документации

При сдаче материалов по данному кейсу предоставляются:

1. Ссылка на удалённый репозиторий в системе контроля версий с полным документированным исходным кодом проекта, историей изменений каждого члена команды, файлами apk и app bundle в разделе Releases;
2. Ссылка на видео с краткой демонстрацией работы приложения длительностью не более 2 минут;

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ
ОЛИМПИАДА ШКОЛЬНИКОВ**
Профиль «Информационные технологии»
Командный кейс № 8«Умный домофон»

3. Техническая документация, включающая титульный лист со списком команды, обоснование выбора архитектуры приложения, структур данных и СУБД, краткое описание используемых средств разработки, UML-диаграммы (class/package/sequence, на выбор), распределение зон ответственности каждого участника команды, описание реализованных алгоритмов и скриншоты приложения (от 6 скриншотов), которые будут в полном объеме отражать проделанную работу (все описанные экранные формы и состояния). Допустимые форматы файла технической документации: .pdf, .docx, .tex.

4. Регламент испытаний

Все члены команды должны зарегистрироваться в системе <https://innovationcampus.ru/lms/>, подать заявку на курс «Московская предпрофессиональная олимпиада» и объединиться в команду.

Загрузить zip-архив с полным проектом Android Studio / IntelliJ IDEA необходимо в систему <https://innovationcampus.ru/lms/>. Для этого перейдите по ссылке <https://innovationcampus.ru/lms/enrol/sic/apply.php?code=VW4HQZXZNO> и зарегистрируйтесь в учебной системе.

После загрузки файла будет произведена распаковка архива, сборка проекта, установка apk на устройства и валидация путём запуска инструментальных тестов. Предварительный рейтинг, учитывающий новое решение команды, будет обновлён в течение 4 часов с момента успешного запуска приложения.

Загружать архив в систему могут несколько членов команды. Будет учтена последняя попытка от команды.

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ

Профиль «Информационные технологии» Командный кейс № 8 «Умный домофон»

После окончания срока сдачи решения кейса для отборочного этапа будет произведена повторная валидация. Оценивание на отборочном этапе учитывает результаты инструментальных тестов, полноту технической документации, ведение git-репозитория, удобство использования приложения, наличие видеодемонстрации. Загрузить исходный код приложения необходимо минимум за 48 часов до завершения практического тура отборочного этапа. В течение 48 ч будет проводиться валидация в Test Lab.

5. Примерный перечень средств и инструментов для выполнения задания

Открытый курс «Мобильная разработка на Java». Samsung Innovation Campus. URL: <https://myitschool.ru/edu/>

Материалы Samsung Android Bootcamp 2020. URL: <https://youtube.com/playlist?list=PLa2T1zmZ6w5IVEYXsemrWOgFShsswVl2A>

Материалы Samsung Android Bootcamp 2021. URL: <https://youtube.com/playlist?list=PLa2T1zmZ6w5KzKoh9M91vk1LBqpc-WtoS>

Марсикано К, Стюарт К, Филлипс Б. *Android. Программирование для профессионалов*. СПб: Питер. 2017.

Murphy, M.L. *The busy coder's guide to Android development*. United States: CommonsWare, 2008.

Murphy, M.L. *Elements of Android Room*. United States: CommonsWare, 2021.

Документация для разработчиков Android приложений. URL: <https://developer.android.com/>

One UI Design Guidelines. URL: <https://developer.samsung.com/one-ui>