МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Математический факультет

Кафедра функционального анализа и операторных уравнений

# Программная генерация задач ЕГЭ по математике (раздел «Алгебра и начала анализа») на языке JavaScript

Бакалаврская работа

Направление 02.03.01 Математика и компьютерные науки

Профиль Математическое и компьютерное моделирование

Зав. кафедрой _____ д. ф.-м. н., проф. М.И. Каменский __.__.20__г.

Обучающийся _____ *С.*Д. Алендарь

Руководитель _____д. ф.-м. н., проф. М.И. Каменский

Воронеж 2024

# Содержание

# Введение

ЕГЭ (Единый государственный экзамен) – это государственный экзамен, проводимый в России для выпускников школ. ЕГЭ проводится по 14 предметам, включая французский, испанский, немецкий и с 2019 года китайский язык. Обязательными являются русский язык и базовый или профильный модуль по математике.

ЕГЭ проходит каждый год в период с мая по июль. Цель экзамена – определить уровень знаний и умений выпускников и принять решение о поступлении в высшие учебные заведения. Результаты ЕГЭ засчитываются как вступительное испытание в вузы России. Сдают ЕГЭ выпускники средних школ, которые получили документ об окончании общеобразовательной школы. В целом, это молодые люди в возрасте 16-18 лет.

Для успешной сдачи ЕГЭ необходима тщательная подготовка, и немаловажную роль в этом играет сайт «Решу ЕГЭ»[5]. Он может быть полезен в подготовке к экзамену по математике. Однако за время подготовки к экзаменам школьники быстро сталкиваются с дефицитом заданий, а в некоторых случаях прибегают к списыванию.

«Час ЕГЭ» — компьютерный образовательный проект, разрабатываемый при математическом факультете ВГУ в рамках «OpenSource кластера» и предназначенный для помощи учащимся старших классов подготовиться к тестовой части единого государственного экзамена.

Задания в «Час ЕГЭ» генерируются случайным образом по специализированным алгоритмам, называемых шаблонами, каждый из которых охватывает множество вариантов соответствующей ему задачи. В настоящее время в проекте полностью реализованы тесты по математике с кратким ответом. Планируется с течением времени включить в проект тесты по другим предметам

школьной программы.

Цель - разработка шаблонов, основанных на задачах из ЕГЭ. Большое и разнообразное количество заданий, генерируемых алгоритмами шаблонов, позволяет объективно оценить знания школьников. Благодаря им решаются проблемы списывания и нехватки заданий.

# 1. Установка «VirtualBox» и «Ubuntu»

1. Перейходим по ссылке https://www.virtualbox.org. Нажимаем кнопку «Download», выбираем «Windows hosts» и устанавливаем VirtualBox.

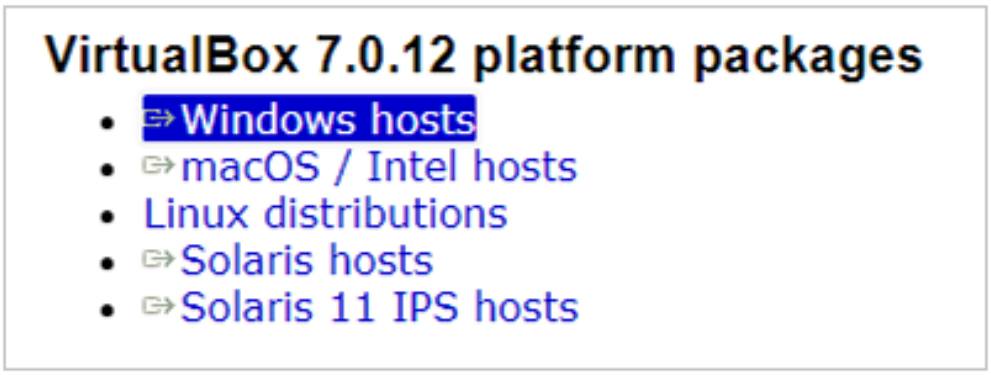


Рис. 1: «Windows hosts».

2. Далее, для того чтобы любая виртуальная машина могла работать, необходимо включить виртуализацию. Чтобы проверить, включена ли она, открываем диспетчер задач и переходим в раздел «Производительность». Открываем окно диспетчера задач на весь экран и внизу увидими, включена ли виртуализация.

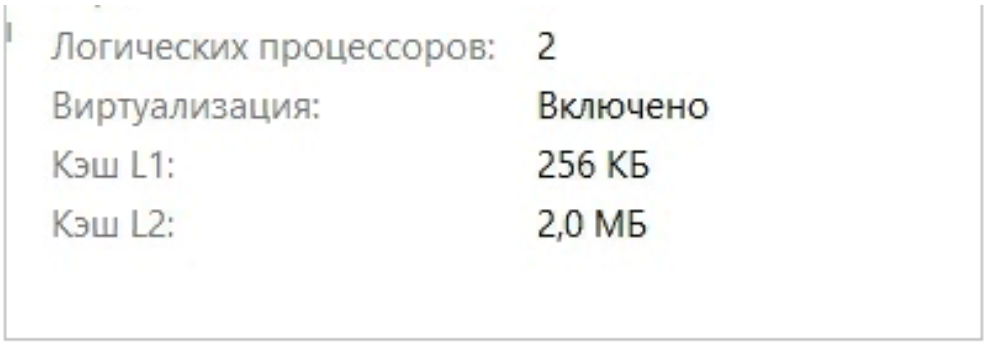


Рис. 2: Диспетчер задач. Виртуализация.

Если виртуализация выключена, то необходимо войти в BIOS. Для этого нужно нажать на кнопку «Перезагрузить компьютер», и, как только он начинает запускаться, зажать клавишу «Esc» ( или начать нажимать много раз на клавишу delete, пока не запустится «Startup menu». Зависит от того, ноутбук у вас или компьютер), после чего и откроется «Startup menu». Чтобы открыть BIOS нажмите клавишу «F10». Переходим в «System Configuration» нажав два раза на кнопку со стрелочкой вправо. Затем спускаемся до «Virtualization Technology», нажав на кнопку со стрелочкой вниз и нажимаем клавишу «Enter». Выбираем «Enable» и снова нажимаем «Enter». Нажимаем клавишу «F10» и выбираем «Yes». После чего компьютер перезагрузится уже с включённой виртуализацией.

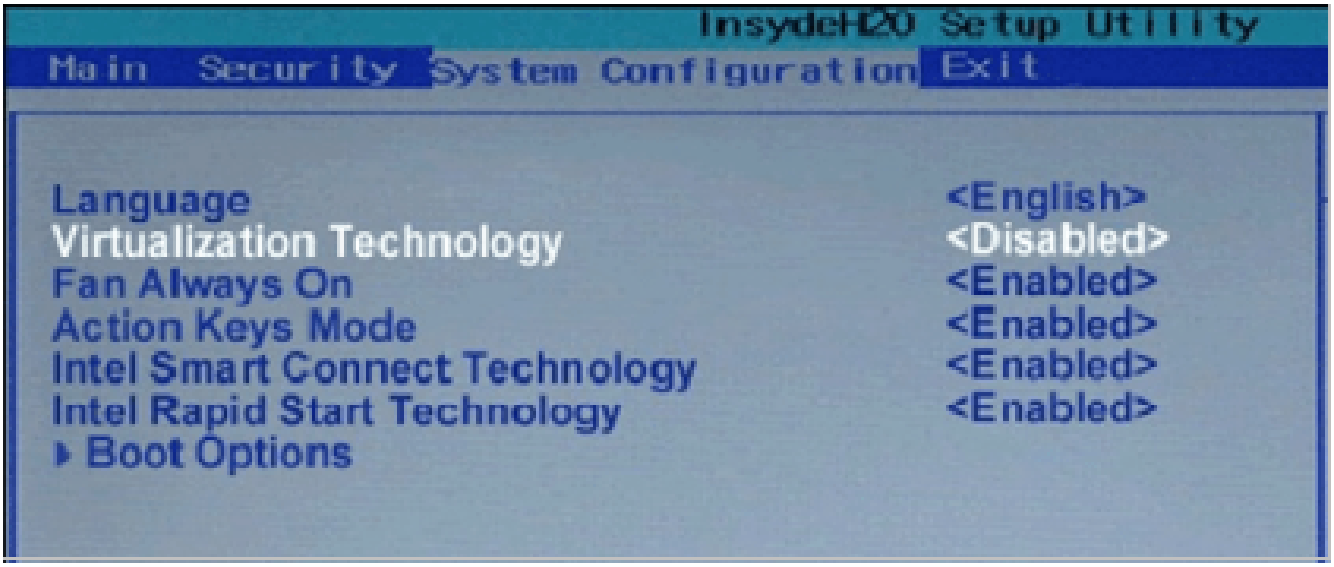


Рис. 3: BIOS. «Virtualization Technology».

3. Скачиваем Ubuntu с официального сайта по ссылке: https://ubuntu.com/download/desktop. Затем открываем VirtualBox и нажимаем кнопку «Создать». Заполняя все поля, необходимо указать объём памяти не менее 2 гигабайт, иначе Ubuntu просто не запустится. В остальном можно принять все установки по умолчанию. После создания, необходимо

нажать на кнопку «Настройки», далее «Носители», у надписи «Контроллер: IDE» нажать на значок диска с плюсом.

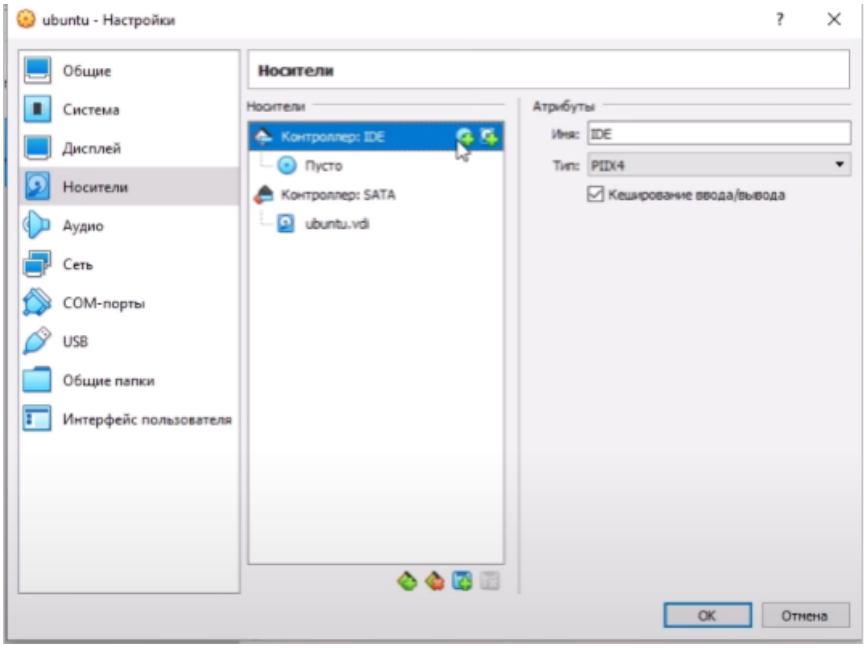


Рис. 4: Настройки. Носители.

Затем «Добавить» и выбрать скаченный ранее файл с Ubuntu. После чего можно нажать кнопку «Запустить». После запуска, мы выбираем язык, нажимаем скачать Ubuntu, заполняем все поля, и выбираем параметры по умолчанию. И наконец видим интерфейс Linux.

# 2. Работа с «github»

1. Установка git, nodejs, npm и grunt.

Для установки всего необходимого, нам нужно открыть терминал, нажав на его иконку.

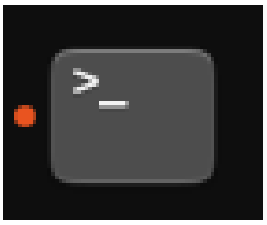

Рис. 5: Терминал.

Далее, для скачивания, необходимо ввести соответствующие строки кода:

`git - sudo apt install git`

`nodejs - sudo apt install nodejs`

`npm - sudo apt install npm`

`grunt - sudo apt install grunt`

Чтобы удостовериться, что всё правильно скачалось, можно узнать версию данного продукта. Например: `git -version`

2. Регистрация на GitHub.

Для регистрации на GitHub нужно перейти по ссылке: : https://github.com/ и заполнить всю необходимую информацию о себе.

3. Работа с репозиторием.

Переходим по ссылке: https://github.com/nickkolok/chas-ege/. Далее нажимаем на зелёную кнопку с надписью «Code», и копируем ссылку репозитория. Лучше сделать это сразу, потому что Ubuntu на VirtualBox может сильно нагружать компьютер, и открыть вкладку с браузером может быть проблематично из-за

нагрузки. (Если возникли проблемы с копирование ссылки, то можно открыть браузер внутри Ubuntu, перейти по ссылке и скопировать ссылку репозитория в нём.)

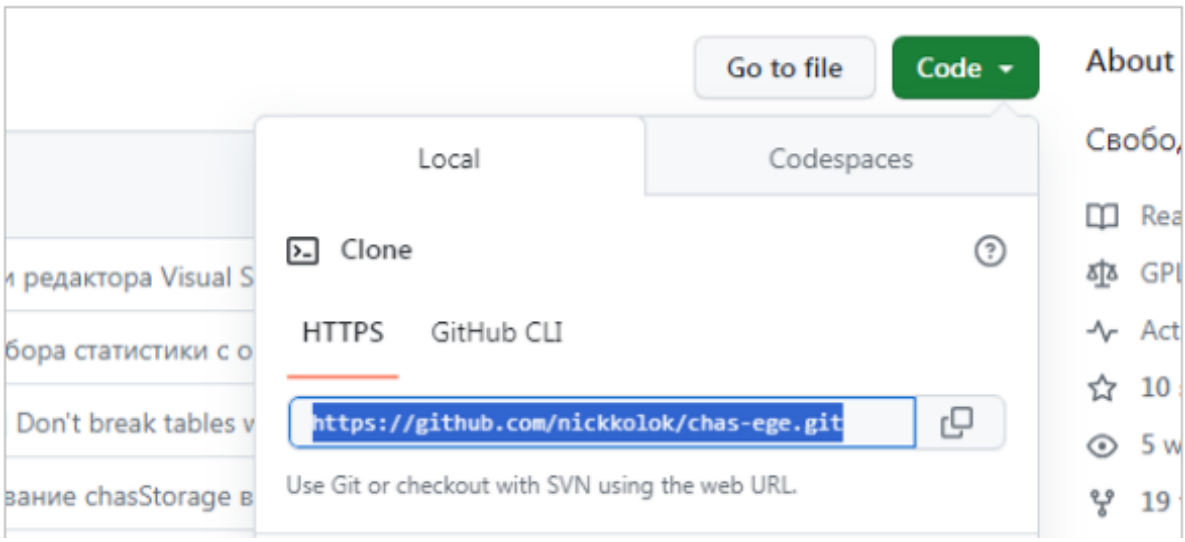


Рис. 6: Github. Ссылка на репозиторий.

• Далее снова заходим в терминал и создаём папку на рабочем столе командой: mkdir <название папки>. Можно убедиться, что папка создана, с помощью команды: ls. Мы увидим все папки на рабочем столе, среди которых должна быть только что созданная.Затем заходим в папку командой: cd <название папки>, и клонируем себе репозиторий командой: git clone <ссылка на репозиторий >

• Добавляем себе ссылку на основной репозиторий проекта с помощью команды: git remote add upstream <ссылка на репозиторий> и убеждаемся, что он подключился, командой: git fetch upstream

• Собираем проект командой: grunt. Важно выполнять эту команду в папке, в которую мы и склонировали репозиторий.

• Открываем файл dist/sh/otladka.html в браузере командой: «open otladka.html», и запускаем любой шаблон, для проверки, в открывшемся окне.

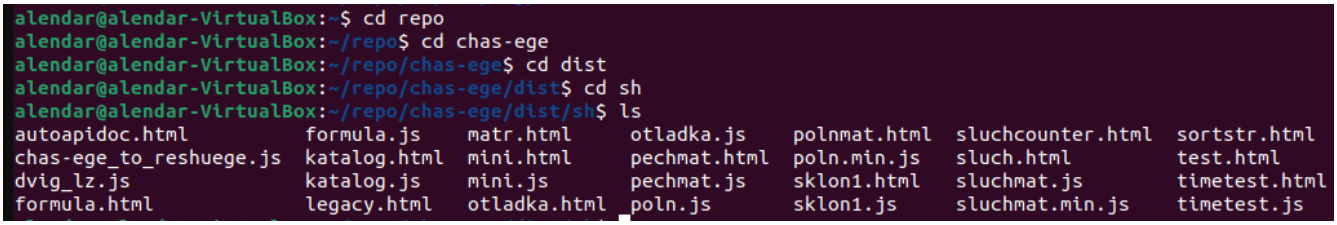


Рис. 7: Путь к файлу otladka.html.

4. Создание SSH ключа для возможности подключиться к удалённому репозиторию. Для начала нужно проверить наличие ключа, введя следующие команды:

```
cd  /.ssh
ls
```

Если файлов с названиями `id_dsa` и `id_dsa.pub` нет (открытый и приватный ключ), то можно создать их используя команду:

```
ssh-keygen -o
```

Далее нужно открыть содержимое файла dsa.pub командой:

```
cat  /.ssh/ id_dsa.pub
```

Далее добавили свой ключ себе в аккаунт на GitHub, чтобы иметь возможность подключиться к удалённому репозиторию. А также прописали ещё одну команду, необходимую для подключения.

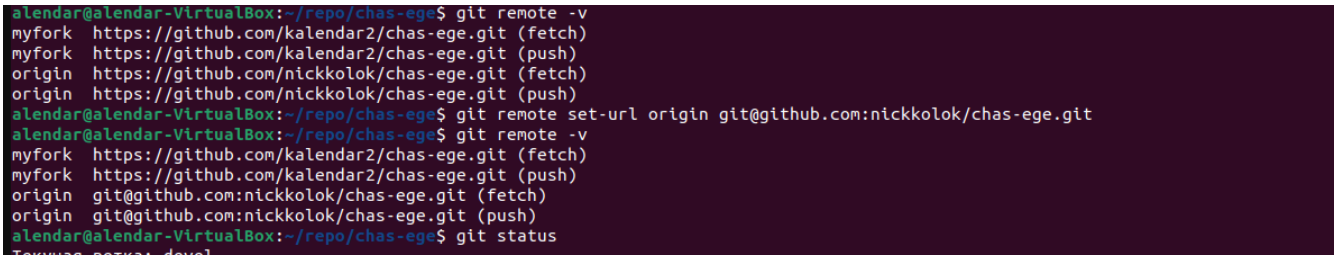


Рис. 8: Подключению к удалённому репозиторию с помощью команды.

Перейдя в репозиторий на GitHub «ЧАС-ЕГЭ», нужно нажать кнопку «Fork» справа вверху.

Открыв VirtualBox и войдя в свою виртуальную машину, открыли терминал и перешли в папку с «Час ЕГЭ» с помошью команды cd git/chas-ege.

Представление гиту выглядит следующим образом:

```
git config -global user.name «Фамилия Имя»
git config -global user.email «электронная почта пользователя»
```

Можно придать выводу гита красные и зелёные цвета с помощью команд: `git config -global color.ui true`

Добавление своего форка на гитхаб в список удалённых репозиториев: `git remote add myfork git@ github.com:«GitHubNik»/chas-ege.git`, где «GitHubNik» - ник пользователя на гитхабе.

Основные команды для создания и отправления изменений в удалённый репозиторий:

Переключение на основную ветку (devel):

```
git checkout devel
```

Её обновление (В некоторых случаях применима также команда: `git pull origin devel`):

```
git fetch origin devel
```

Создание новой ветки:

```
git checkout -b newtask-777
```

Проверка изменений, а также самой ветки:

```
git status
```

Добавление всех изменений:

```
git add .
```

Добавление всех изменений:

```
git commit -m «Внесены изменения в файл ...»
```

Отправка изменений в удалённый репозиторий:

```
git push myfork newtask-777:myfork newtask-777:
```

5. Игнорирование ненужных файлов и каталогов.

Для того, чтобы сообщить Git, какие файлы или каталоги нужно игнорировать , можно создать .gitignore файл.

Для этого необходимо открыть Терминал и перейти к расположению репозитория Git. Далее создаётся файл для репозитория, с помощью команды `touch .gitignore`.

Если файл уже был отправлен в репозиторий, необходимо отменить отслеживание файла, прежде чем добавлено правило игнорирования, с помощью команды: `git rm -cached FILENAME`.

Файл не видим, как и все файлы с точкой в начале названия файла, но его можно увидеть с помощью команды `ls -a`. Зайдя в него, нужно записать путь к файлу или каталогу, который будет игнорироваться.

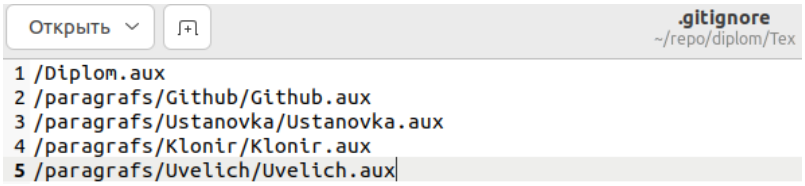


Рис. 9: Файл .gitignore с списком файлов для игнорирования в нём.

Это не единственный способ задания игнорирования файла или каталога. Можно также сообщить Git всегда игнорировать определенные файлы во всех репозиториях на компьютере. Для этого, например, каталог нужно добавить в файл с именем ignore , расположенным внутри каталога `/.config/git`.

Также можно вообще не создавать файл .gitignore. Этот метод можно использовать для локально создаваемых файлов, которые не должны создавать другие пользователи. Для этого, используя текстовый редактор, нужно открыть файл, вызываемый `.git/info/exclude` в корневом каталоге репозитория Git.

# 3. Клонирование виртуальной машины

1. В ходе работы с виртуальной машиной возникла необходимость перенести её с одного своего устройства на другое.

Для этого открываем Virtual Box, нажимаем правой кнопкой мыши по своей виртуальной машине и выбираем пункт клонировать. В появившемся окне указываем имя нового клона и его путь по которому он будет сохранён. Так же в графе «Политика MAC-адреса» выбираем вариант: «Сгенерировать новые MAC-адреса всех сетевых адаптеров».

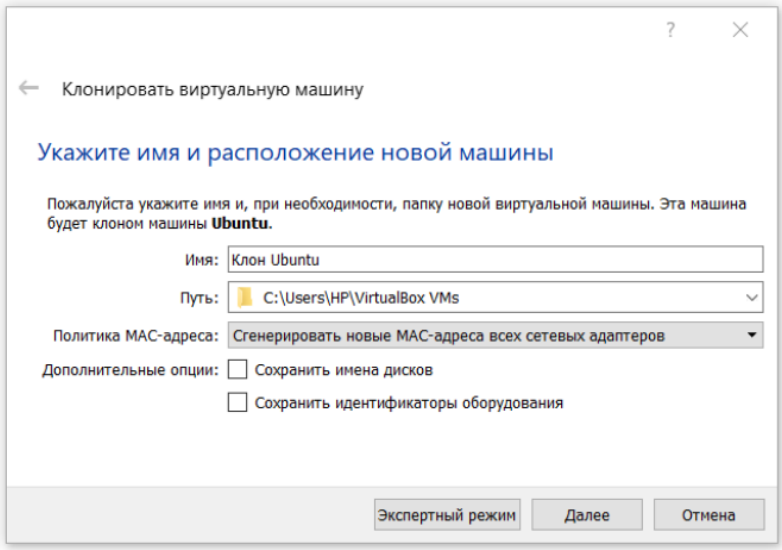


Рис. 10: Окно клонирования.

В следующем окне нужно было указать тип клонирования: полное или связное. При связном клонировании будет создана новая машина, использующая файлы виртуальных жёстких дисков клонируемой машины и нельзя перенести её на другой компьютер без переноса клонируемой. При полном клонировании, будет создана полная копия клонируемой виртуальной машины (включая все файлы виртуальных жёстких дисков). Поэтому выбиираем полное клонирование.

В окне с указанием цели клонирования, указываем клонировать всё, чтобы новая машина не только отражала текущее состояние клонируемой машины, но и имела копии всех снимков её древа снимков.

Далее нажимаем на кнопку «клонировать», после чего запускается процесс клонирования. По его завершению переносим новую машину на флэшку. Это можно сделать нажав в Virtual Box на клон правой кнопкой мыши и выбрать пункт «Переместить». Или просто зайти в папку, в которую был сохранён наш клон, и переместить его уже оттуда.

Далее подсоединяем флэшку к другому компьютеру и переносим машину в папку Virtual Box. Открыв Virtual Box, нажиимаем вверху на кнопку «Машина» и выбрал пункт добавить. После чего находим свою машину и нажал кнопку «Открыть».

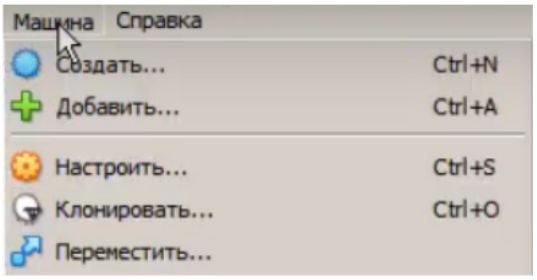


Рис. 11: Добавление виртуальной машины.

Виртуальная машина добавлена. Но зайдя в настройки, можно увидеть, что объём выделенной основной памяти составляет всего лишь 2 гигабайта, что слишком мало для работы с машиной. Так как наша машина находится в состоянии «Сохранена», мы не можем изменять её настройки. Поэтому нажимаем правой кнопкой мыши по перенесённому клону, и выбираем пункт «Сбросить сохранённое состояние». После чего снова нажимаем правой кнопкой мыши по машине, выбираем пункт «Настроить. . . » и в «Системе» выделяем нужное количество памяти.

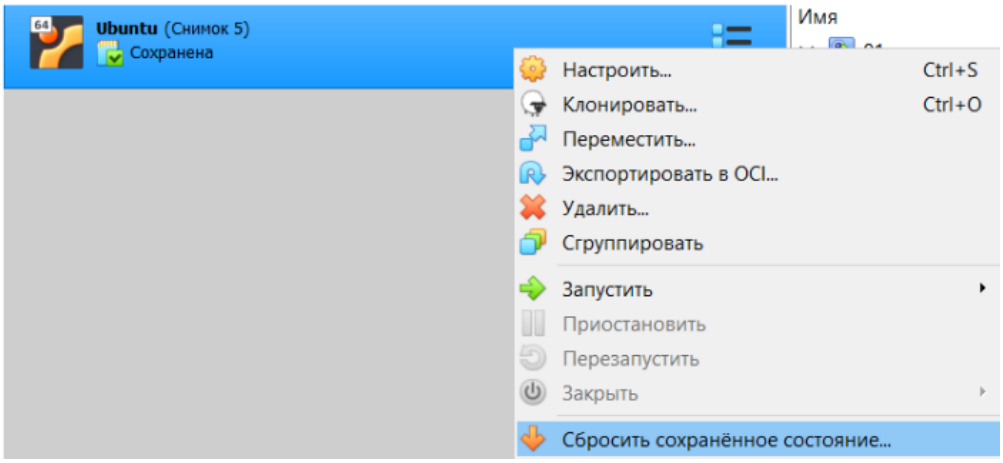


Рис. 12: Настройка памяти.

Далее заходим в «Носители» и выбираем свой жёсткий диск, так как иначе при запуске виртуальной машины мы бы ничего не увидели.

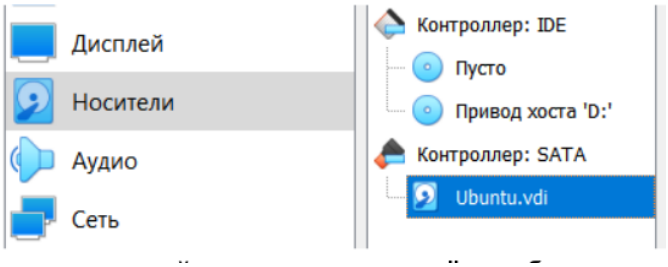


Рис. 13: Настройки. Носители.

Теперь клон виртуальной машины перемещён, добавлен на новый компьютер и с ним можно работать.

Есть и альтернативный способ переноса виртуальной машины с одного устройства на другое, с помощью функций «Экспорт» и «Импорт».

Экспорт виртуальной машины

Экспорт конфигурации виртуальной машины происходит в файл формата .ova (Open Virtual Appliance). Это универсальный формат для хранения данных виртуальной машины, файлы .ova могут использоваться в разных программах виртуализации: VirtualBox, VMware Workstation, Microsoft Hyper-V.

Виртуальная машина, экспортированная в файл .ova, затем может быть импортирована как в VirtualBox, так и в VMware Workstation, Microsoft Hyper-V.

В меню программы нужно зайти в «Файл» и выбрать пункт «Экспорт конфигураций». В открывшемся окне выбираем машину для экспорта, и нажимаем «Далее».

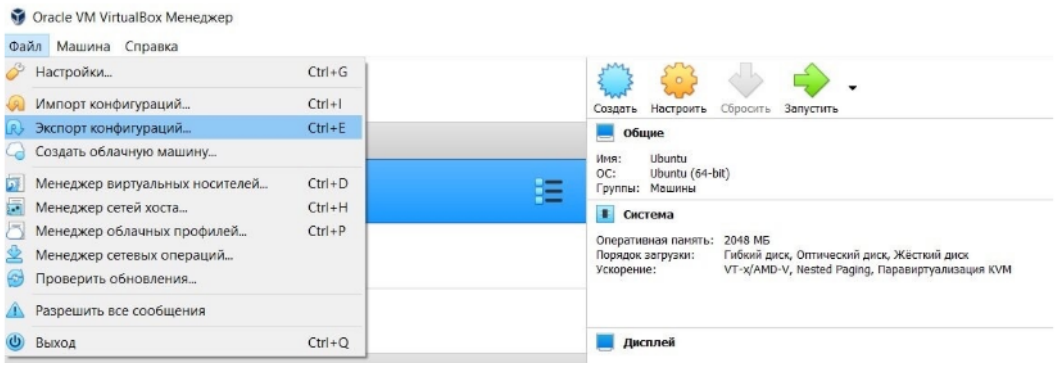


Рис. 14: «Экспорт конфигураций».

Выбираем место размещения после экспорта. Также лучше выбрать «Включать MAC-адреса всех сетевых адаптеров», нажимаем «Далее».

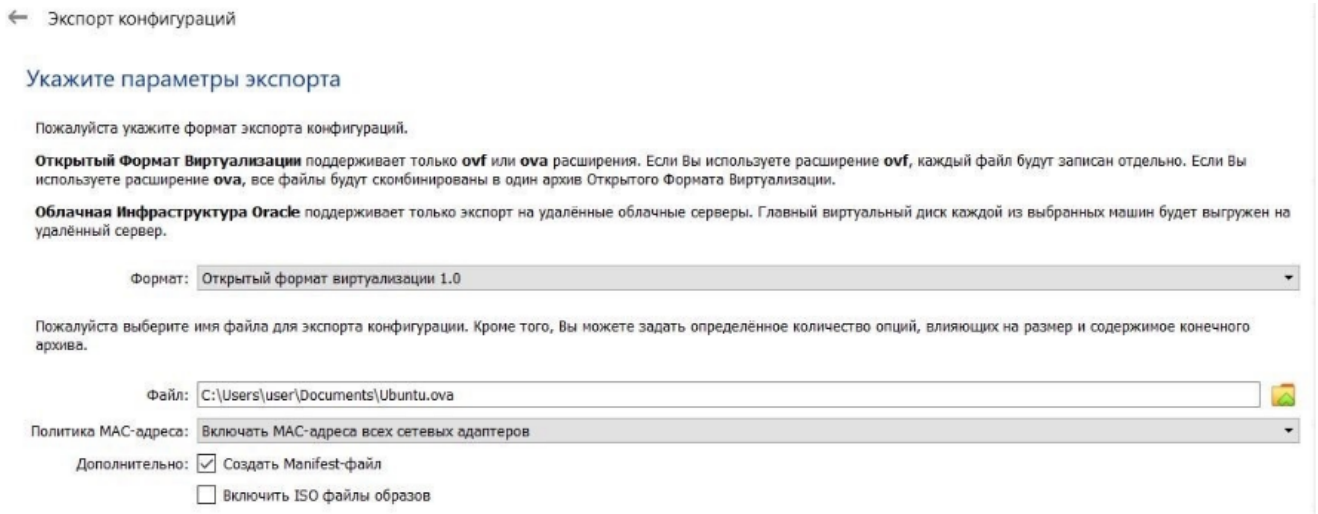


Рис. 15: MAC-адреса сетевых адаптеров.

В следующем окне оставляем всё без изменений, и нажимаем кнопку “Экспорт”. Сам экспорт может занимать несколько минут, в зависимости от раз-

мера виртуальной машины. После экспорта в указанном месте создается файл, который уже необходимо будет импортировать.

Импорт виртуальной машины

Теперь необходимо скопировать файл на флэшку. Или же можно воспользоваться облочным хранилищем, так как возможно при попытке перенести его на флешку, может возникнуть ошибка: «Файл слишком велик для конечной файловой системы». Она возникает, если передаётся файл размером более четырёх гигабайт на носитель, неспособный с ним работать. Для устранения этой ошибки можно воспользоваться форматированием флешки или разбитием файла с виртуальной машины на несколько частей. Был также опробован способ сжатия виртуальной машины в zip файл, но она оказалась практически без сжимающий составляющих.

Далее на втором компьютере заходим в программу Virtual Box и нажимаем вверху «Файл» и выбираем пункт «Импорт конфигураций».


Рис. 16: Импорт конфигураций.

В окне импорта выбираем место размещения файла виртуальной машины, нажимаем «Далее». В следующем окне можно изменить параметры импор-

та, например, увеличить количество процессоров. Также желательно «Включать (сгенерировать новые) MAC-адреса всех сетевых адаптеров», и нажимаем «Импорт».

Импорт также как и экспорт в зависимости от размера виртуальной машины может занимать несколько минут.

После импорта виртуальная машина появляется в списке и с ней уже можно будет работать.

# 4. Увеличение объёма памяти виртуальной машин

В какой-то момент при работе за виртуальной машиной, высветилось сообщение о том, что память заканчивается.

Виртуальные жесткие диски — это форматы файлов образа диска, которые имеют функциональные возможности, как и у физического жесткого диска. В первую очередь они предназначены для использования с виртуальными машинами. Как и физический диск, виртуальный носитель имеет размер и ёмкость, которые указывают при создании диска. Но в отличие от физического носителя виртуальный можно увеличивать.

Для начала вводим в поисковике Linux-а «Диски» и, нажав на иконку диска, смотрим насколько он заполнен, и сколько памяти осталось.

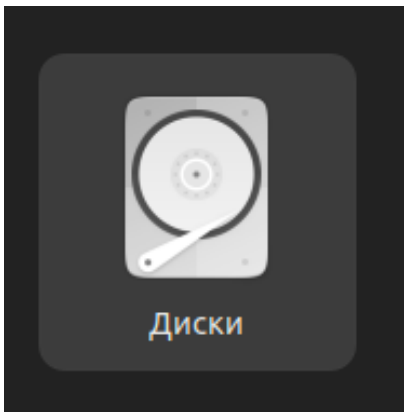


Рис. 17: Иконка диска.

Также количество оставшейся на диске памяти можно проверить, введя в терминале команду: `df`. Но по умолчанию она выводит информацию в байтах. Для более простой оценки свободного места, можно использовать команду: `df -h`. Она уже выводит размеры в гигабайтах, мегабайтах и килобайтах.

Необходимо выключить виртуальную машину, а не сохранить, чтобы можно было изменить её настройки. Далее в меню VirtualBox выбираем пункт

«Файл» и нажимаем на «Менеджер виртуальных носителей...».

В открывшемся окне заходим в свойства диска и находим его размер. Двигая за ползунок, изменяем его объём с 16 гигабайт до 33 и нажимаем «Применить».

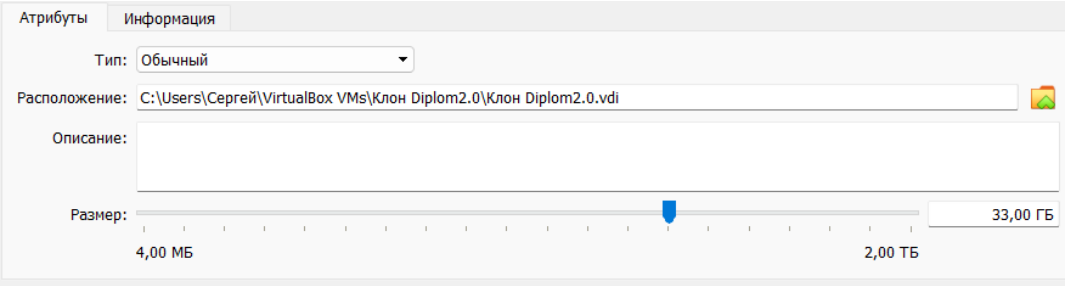


Рис. 18: Изменение размера диска.

При повторном открытии виртуальной машины было обнаружено, что размер памяти не изменился. Мы пришли к выводу, что для того чтобы увеличить объём памяти жёсткого диска, нужно сделать клон текущего состояния виртуальной машины или удалить снимки. К сожалению изменять размер диска, просто нельзя, пока существуют его точки сохранения. Поэтому снова выключаем машину, и делаем клон её текущего состояния. Он делается точно так же, как когда мы делали клон для переноса его с одного устройства на другое. Единственное отличие в том, что в этот раз выбираем клонировать не всё, а только текущее состояние.

Увеличив в «Менеджере виртуальных носителей...» уже объём памяти клона, заходим в настройки виртуальной машины, увеличиваем размер памяти в «Системе» и выбраем соответствующий диск в «Носителях», так же как это делали, после переноса клона на другое устройство. Далее запускаем новую машину и заходим в «Диск», где видим, что его размер стал больше, и появилась дополнительная область памяти.

Чтобы задействовать его, нажимаем на иконку настроек, сдвигаем пол-

зунок до конца вправо, чтобы можно было использовать всю новую память и нажиимаем «Применить», после чего вводим свой пароль.

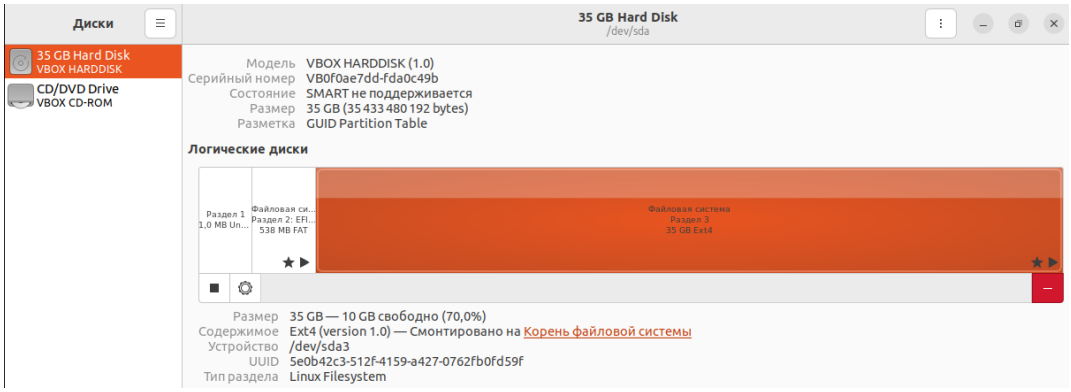


Рис. 19: Увеличенный диск.

Но опять ничего не получилось. Чтобы наконец всё заработало, нужен установочный файл Ubuntu нашей виртуальной машины. Он остался на прошлом устройстве, с которого мы и клонировали машину. Воспользовавшись гугл диском, перенесли файл, так как флэшка не позволяет загрузить в неё что-либо весом более 4 гигабайт без её форматирования.

После перенесения установочный файла на новое устройство, необходимо добавить его в настройках виртуальной машины. Для этого нажать «Настроить», зайти в «Носители», кликнуть по пустому диску, и в «Оптическом приводе» выбрать файл установки.

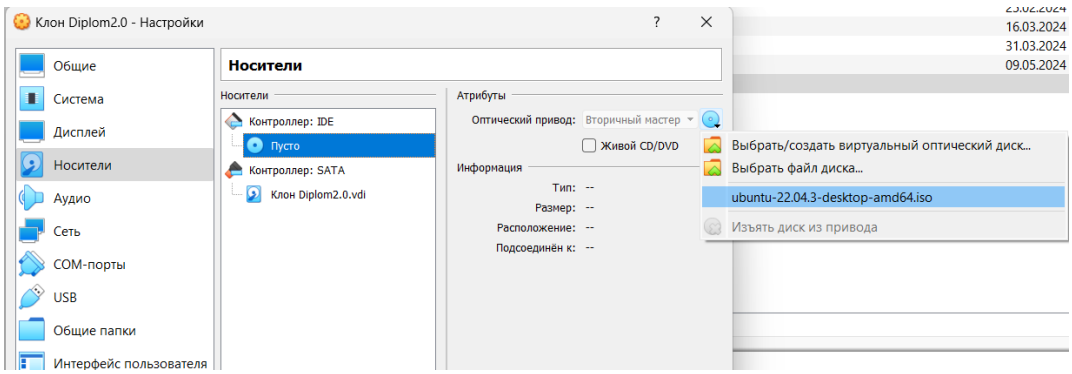


Рис. 20: Добавление установочного файла.

При запуске виртуальной машины, нужно выбрать пункт "Try or Install Ubuntu"и далее нажать "Try Ubuntu".

Необходимо так же установить приложение GParted.

• Первым шагом нужно обновить списки пакетов, введя в терминале команду: «sudo apt-get update».

• Далее установить GParted командой: `sudo apt-get install gparted`.

• Ввести пароль, когда это предложат. Сам пароль и то как он вводится будет не видно.

• После установки GParted, его наконец-то можно запустить из меню приложений или с помощью команды: `sudo gparte`.

В открывшемся меню нужно кликнуть правой кнопкой мыши по области `/dev/sda3` и выбрать пункт «Изменить размер/Переместить». Далее, двигая за ползунок с правого края, можно наконец позволить виртуальной машине использовать всё добавленное, свободное пространство. Снова нажимаем на кнопку «Изменить размер/Переместить». И в меню GParted кликаем по зелёной галочке, чтобы применить все внесённые изменения. После недолгой загрузки закрываем GParted.

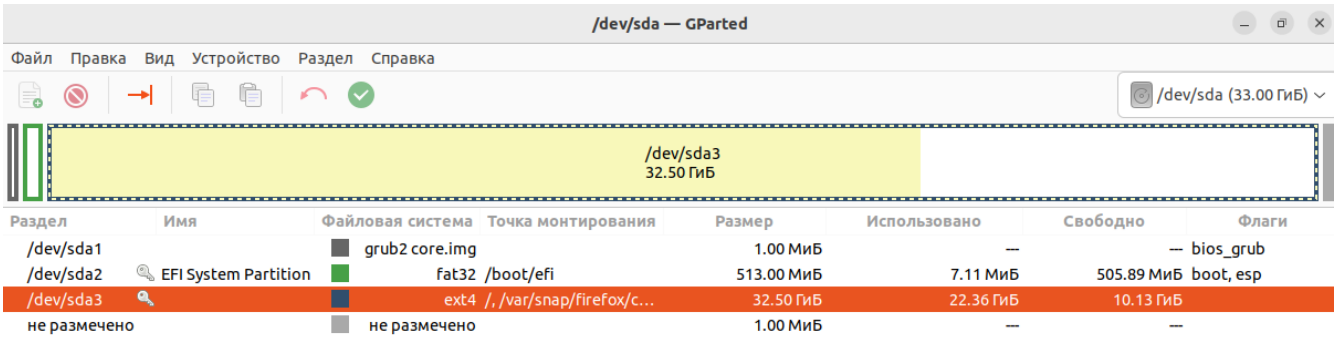


Рис. 21: Gparted.

Последний раз убеждаемся, что размер диска увеличен и свободное пространство используется, с помощью команды: `df -h`.

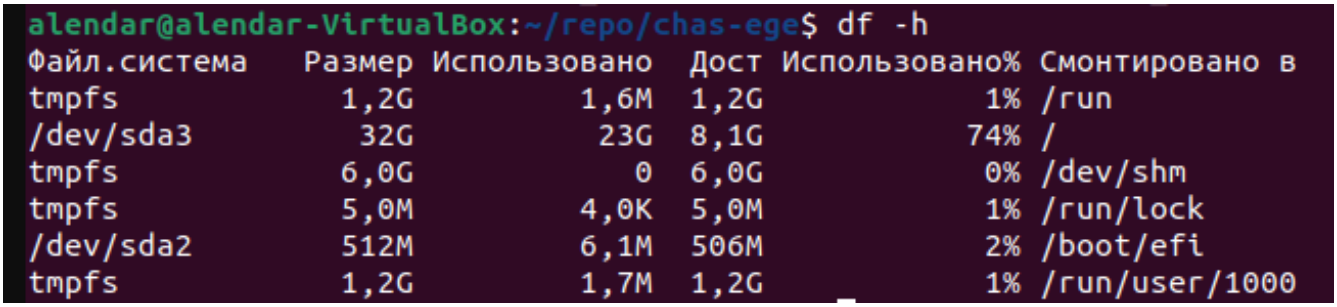


Рис. 22: Вывод команды `df -h`.

Чтобы при каждом запуске виртуальной машины, установочный файл не предлагал установить Ubuntu, нужно удалить его в настройках виртуальной машины так же, как мы его и добавляли.

# Примеры задач и их код

Пример пары заданий одного типа, на основе которых создан следующий шаблон.

**Задание 3.** Материальная точка движется прямолинейно по закону $x(t)=\frac{1}{3}t^3-5t^2+45t+130$, где $x$ – расстояние от точки отсчёта в метрах, t – время в секундах, измеренное с момента начала движения. В какой момент времени её скорость была равна 20 м/с?

**Задание 6.** Материальная точка движется прямолинейно по закону $x(t)=\frac{1}{6}t^3-2t^2+6t+250$, где $x$ – расстояние от точки отсчёта в метрах, t – время в секундах, измеренное с момента начала движения. В какой момент времени её скорость была равна 96 м/с?

Первая часть шаблона выглядит следующим образом:

```
(function() {
        'use strict';
        var a = sluchch(0.5, 1, 0.5);
        var b = 0;

        if (a == 0.5) {
            b = 6;
        } else {
            b = 3;
        }

        function plusmin(member) {
            var el = math.parse(member);
            return math.simplify(el).toTex();
        }

        var x2 = sluchch(-10, -1);
        var x1 = Math.abs(x2) + sluchch(3, 10);
        var c = Math.abs(a * x1 * x2) + sluchch(1, 20);
        var d = sluchch(1, 999);
```

Здесь объявляются переменные таким образом, чтобы генерируемые значения, максимально соотносились с исходной задачей.

В этой части также используются специальные встроенные функции, такие как:

`sluchch(a, b)` – функция случайным образом возвращает число из диапазона от «a» до «b». Если функции добавить третье число `sluchch(a, b, c)`, то она будет возвращать случайно число уже с шагом «c».

`Math.abs(a)` – функция возвращает модуль числа «a».

`function plusmin(member)` – функция принимает аргумент, содержащий в себе пример, наподобие: «1*a+-b». И возвращает его упрощённый вариант: «a-b».

Вторая часть выглядит следующим образом:

```
21    chas2.task.setTask({
22        text: 'Материальная точка движется прямолинейно по закону:$$ x(t)=' +
23            '\\frac{1}{' + b + '}' + plusmin('x^3+' + (((-1) * a *
24                (x1 + x2)) / 2)) + plusmin('x^2+' + (a * x1 * x2 +
25                c)) + 'x+' + d + ',' + '$$' +
26        'где $x$ - расстояние от точки отсчёта в метрах, $t$ -' +
27        'время в секундах, измеренное с момента начала движения.' +
28        'В какой момент времени её скорость была равна ' + c + ' м/с?',
29        analys: '$$' + 'Решение:' + '$$' +
30            'Найдём закон изменения скорости: ' + '$$' +
31            'v(t)=x^{\'}(t)=' + plusmin(a + 'x^2' + (
32                (-1) * a * (x1 + x2))) + 'x' + '+' + (a * x1 * x2 +c) + '$$' +
33            'Чтобы найти, в какой момент времени $t$ скорость была равна ' +
34            c + ' м/с, решим уравнение:' + '$$' + plusmin(a + 'x^2' +
35                ((-1) * a * (x1 + x2))) + 'x' + '+' + (a * x1 * x2 +
36                c) + ' = ' + c + ' \\Rightarrow\\ ' + plusmin(a +
37                'x^2' + ((-1) * a * (x1 + x2))) + 'x' + plusmin('+' + (a * x1 * x2)) +
38            ' = 0' + '$$' + '$$' + 'D = b^2-4ac = ' + ((-1) * a * (x1 + x2)) +
39            '^2-4\\cdot' + a * a * x1 * x2 + '=' + (Math.pow(((-1) *
40                a * (x1 + x2)), 2) - (4 * a * a * x1 * x2)) + '$$' +
41            '$$\\begin{cases}' + 'x_1=' + '\\frac{-b+\\sqrt{D}}{2a}' + '=' +
42            '\\frac{' + a * (x1 + x2) + '+\\sqrt{' + (Math.pow(((-1) * a * (x1 + x2)),
43                2) - (4 * a * a * x1 * x2)) + '}}{' + 2 * a + '}' +
44            ' = ' + x1 + '\\\\~\\\\' + 'x_2=' +
45            '\\frac{-b-\\sqrt{D}}{2a}' + '=' + '\\frac{' + a * (x1 +
46                x2) + '-\\sqrt{' + (Math.pow(((-1) * a * (x1 + x2)),
47                2) - (4 * a * a * x1 * x2)) + '}}{' + 2 * a + '}' +
48            ' = ' + x2 + '\\end{cases}$$',
49        answers: x1,
50    });
51})();
52//119919
```

Вторая часть программы делится на несколько составных частей:

условие задачи, записанное после `text:`

решение, записанное после `analys:`

ответ, записанный после `answer`

В этом шаблоне мы вводим переменные, генерирующие случайные значения, которые используем для составления ответа. И на основе ответа мы уже

составляем саму задачу и её решение.

В этой части также используются некоторый команды latex, например:

`\frac` – функция выводящая дробь на экран.

`\Rightarrow` – функция выводящая двойную стрелку вправо

`\begin{cases}` и `\end{cases}` – окружение, выводящая фигурную скобку на экран.

Пример сгенерированных шаблоном задач

Материальная точка движется прямолинейно по закону:

$$x(t) = \frac{1}{6}x^3 + \frac{-7}{2}x^2 - 76x - 796,$$

где $x$ - расстояние от точки отсчёта в метрах, $t$ - время в секундах, измеренное с момента начала движения.
В какой момент времени её скорость была равна $-40$ м/с?

Найдём закон изменения скорости:

$$v(t) = x'(t) = \frac{x^2}{2} - 7x - 76$$

Чтобы найти, в какой момент времени $t$ скорость была равна $-40$ м/с, решим уравнение:

$$\frac{x^2}{2} - 7x - 76 = -40 \Rightarrow \frac{x^2}{2} - 7x - 36 = 0$$

$$D = b^2 - 4ac = -7^2 + 4 \cdot 18 = 121$$

$$\begin{cases} x_1 = \dfrac{-b + \sqrt{D}}{2a} = \dfrac{7 + \sqrt{121}}{1} = 18 \\ x_2 = \dfrac{-b - \sqrt{D}}{2a} = \dfrac{7 - \sqrt{121}}{1} = -4 \end{cases}$$

Материальная точка движется прямолинейно по закону:

$$x(t) = \frac{1}{6}x^3 + \frac{9}{2}x^2 + 40x - 721,$$

где $x$ - расстояние от точки отсчёта в метрах, $t$ - время в секундах, измеренное с момента начала движения.
В какой момент времени её скорость была равна 60 м/с?

Найдём закон изменения скорости:

$$v(t) = x'(t) = \frac{x^2}{2} + 9x + 40$$

Чтобы найти, в какой момент времени $t$ скорость была равна 60 м/с, решим уравнение:

$$\frac{x^2}{2} + 9x + 40 = 60 \Rightarrow \frac{x^2}{2} + 9x - 20 = 0$$

$$D = b^2 - 4ac = 9^2 + 4 \cdot 10 = 121$$

$$\begin{cases} x_1 = \dfrac{-b + \sqrt{D}}{2a} = \dfrac{-9 + \sqrt{121}}{1} = -20 \\ x_2 = \dfrac{-b - \sqrt{D}}{2a} = \dfrac{-9 - \sqrt{121}}{1} = 2 \end{cases}$$

Задачи из ЕГЭ по математике базового уровня тип №16

1) $(0{,}01)^2 \cdot 10^4 : 3^{-2}$

2) $(0{,}01)^2 \cdot 10^5 : 4^{-2}$

3) $(0{,}1)^2 \cdot 10^4 : 2^{-3}$

4) $(0{,}1)^5 \cdot 10^3 : 7^{-2}$

5) $(0{,}1)^4 \cdot 10^3 : 5^{-2}$

$$\frac{\log_5 12^{72}}{8 \log_5 12}$$

Шаблоны задач

```
(function() {
    retryWhileError(function() {
        'use strict';
        let a = sl(2, 19);
        let b = sl(2, 9);
        let d = sl(2, 9);

        NAtask.setEvaluationTask({
            expr: '(varlog(' + b + ',' + a + '^' + d * sl(1, 9) + '))/(' + d +
            '(varlog(' + b + ',' + a + ')))',
            authors: ['Алендарь Сергей'],
        });
    }, 10000);
})();
//10001803

(function() {
    retryWhileError(function() {
        'use strict';
        NAtask.setEvaluationTask({
            expr: 'forceBrackets(' + [0.01, 0.1].iz() + ')^' + sl(2, 6) + '*' +
                'divideColon(10^' + sl(2, 6) + ',' +
                sl(2, 9) + ' ^ ' + sl(-1, -5) + ')',
            authors: ['Алендарь Сергей'],
        });
    }, 10000);
})();
//509707
```

Отличительной особенностью этих шаблонов от предыдущего, является другая библиотека - `setEvaluationTask`. Помимо самой генерации задачи, на основе переменных их первой части, она так же сама решает сгенерированный пример, и проверяет его ответ.

Если ответ не является удовлетворительным, а именно таким, что пользователь просто не сможет ввести его на клавиатуре (Ответ представляет из себя

бесконечную десятичную дробь, или просто очень длинную), то программа, отслеживает эту ошибку с помощью цикла `retryWhileError` и предпринимает ещё попытку составить задание с цельным ответом. Количество таких попыток указано внизу кода программы. Если за установленное число попыток, программа так и не получит правильно сгенерированную задачу, то она выведет сообщение об ошибке.

В этих шаблонах тоже используются некотрые свои специальные встроенные функции, например:

`varlog` - отвечает за отображение логарифма на экране.

Также эта библиотека способна сама убирать некоторые скобки, если они не влияют на решение задания, или сама выполнть арифметическую операцию. Но в некотрых случаях, это не является необходимостью. Чтобы программа не высчитывала сама, то что стоит оставить, используются следующие команды:

`forceBrackets` - для сохранение скобок.

`divideColon` - для вывода знака деления «:».

Сгенерированные по шаблонам задания

Найдите значение выражения:

$$\frac{\log_3 8^{56}}{7\ \log_3 8}$$

Найдите значение выражения:

$$\frac{\log_5 9^{15}}{3\ \log_5 9}$$

Найдите значение выражения:

$$(0{,}1)^4 \cdot 10^3 : 6^{-4}$$

Найдите значение выражения:

$$(0{,}01)^3 \cdot 10^4 : 8^{-4}$$

Задача №77381 тип №6

Найдите корень уравнения

$$-2 + \log_3 (42 + 21x) = \log_3 (6 + 2x)$$

Решение:

$$\log_3 (42 + 21x) = \log_3 (6 + 2x) + 2$$

$$2 = \log_3 3^2, \ \log_a b + \log_a c = \log_a bc \ \Rightarrow \ \log_3 (6 + 2x) + \log_3 9 = \log_3 ((6 + 2x) \cdot 9) \ \Rightarrow$$

$$\Rightarrow \begin{cases} 42 + 21x > 0 \\ 6 + 2x > 0 \\ (42 + 21x) = (6 + 2x) \cdot 9 \end{cases} \Rightarrow \begin{cases} 21x > -42 \\ 2x > -6 \\ 42 + 21x = 54 + 18x \end{cases} \Rightarrow \begin{cases} x > \dfrac{-42}{21} \\ x > \dfrac{-6}{2} \\ 3x = 12 \end{cases} \Rightarrow \begin{cases} x > -2 \\ x = 4 \end{cases}$$

Ответ: $x = 4$.

Первая часть шаблона

Вторая часть шаблона

Задачи, сгенерированные по шаблонам, использующим код арифметической прогрессии.

Код арифметической прогрессии

Задачи, сгенерированные по шаблонам, использующим код геометрической прогрессии.

Код геометрической прогрессии

Тригонометрическая задача №77492

Код для задачи №77492

# Заключение

За всё время работы было разработано следующее количество шаблонов:

Тип №10 по арифметической прогрессии: 9

Тип №10 по геометрической прогрессии: 4

Тип №6 ЕГЭ профильного уровня: 5

Тип №8 ЕГЭ профильного уровня: 3

Тип №12 ЕГЭ профильного уровня: 12

Тип №13 ЕГЭ профильного уровня: 1

Тип №16 ЕГЭ базового уровня: 29

Также были подготовлены тезисы, которые были отправлены для заочного участия в 74 МСНТК.

Данная работа помогла оказать вклад в развитии образовательного проекта "Час ЕГЭ а разработанные шаблоны, помогут школьникам в подготовке к тестовой части ЕГЭ по математике.

Помимо этого, сама дипломная работа, будет служить руководством, для будущих студентов, которые будут также принимать участие в разработке шаблонов и развитии проекта "Час ЕГЭ".

# Список Литературы

1. Полный интерактивный тест - Тренажёр «Час ЕГЭ» - URL: https://math.vsu.ru/chas-ege/sh/polnmat.html (дата обращения: 18.04.2024).

2. Львовский С. М. «Набор и верстка в системе LaTeX» - URL: https://old.mccme.ru//free-books//llang/newllang.pdf (дата обращения: 16.04.2024).

3. Полноценная платформа для разработчиков для создания, масштабирования и доставки защищенного программного обеспечения GitHub — URL: https://github.com/nickkolok/chas-ege/ (дата обращения: 20.04.2024).

4. Открытый банк задач ЕГЭ по Математике. Профильный уровень. – URL: https://prof.mathege.ru

5. Образовательный портал «РЕШУ ЕГЭ» - URL: https://ege.sdamgia.ru

```
(function() {
    retryWhileError(function() {
        'use strict';

        var v = sl1();
        var d = v ? '+' : '-';
        var znak = v ? 1 : (-1);

        var a = sluchch(2, 3);
        var b;
        if (a === 2) {
            b = znak * sluchch(4, 64, 8).pm();
        } else {
            b = sluchch(15, 69, 27);
        }

        var y = sluchch(1, 3);
        var c = sluchch(1, 3);
        var t = -znak * (12 + b) / a.pow(c);
        var u = sluchch(1, 4);
        var z = y * a.pow(c) + u;

        var x = (t * a.pow(c) - b) / (z.plusminus() - y.plusminus() * a.pow(c));
        genAssertZ1000(x, "Корень очень нецелый");

        var x1 = -b.plusminus() / z;
        var x2 = -t.plusminus() / y;
        genAssertZ1000(x1, "Корень очень нецелый");
        genAssertZ1000(x2, "Корень очень нецелый");
        genAssert(x > -b.plusminus() / z, "Корень не попадает в ОДЗ");
        genAssert(x > -t.plusminus() / y, "Корень не попадает в ОДЗ");

        var ans = x1;
        var ans1;
        var q = b;
        var j = z;

        if (x2 > x1) {
            ans = x2;
            q = t;
            j = y;
        }

        if (ans.isZ()) { //Дробная часть для смешанного числа
            ans1 = ' ';
        } else {
            ans = Math.trunc(ans);
            ans1 = '\\frac{' + (q.abs() % j) + '}{' + j + '}';
            if (ans === 0) {
                if (-q.plusminus() < 0) {
                    ans = '-';
                } else ans = ' ';
            }
        }

```

```
        chas2.task.setAdditiveEquationTask({
            parts: ['-\\log_{' + a + '}{(' + [b.plusminus(), z + 'x'].slag() +
             ')}', '\\log_{' + a + '}{(' + [t.plusminus(),
                y + 'x'].slag() + ')}', c],
            roots: x,
            enablePartsSubtraction: [1, 0],
        }, {
            tags: {
                log: 1,
            },
            analys: 'Решение:' +
                '$$\\log_{' + a + '}{(' + b.plusminus() + '+' + z + 'x' + ')}=' +
                 '\\log_{' + a + '}{(' + t.plusminus() + '+' +
                y + 'x' + ')}+{' + c + '}$$' +
                '$${' + c + '} =\\log_{' + a + '}{' + a + '^' + c + '},\\
                 \\log_{a}{b}+' + '\\log_{a}{c}=' +
                '\\log_{a}{bc} \\ \\Rightarrow\\ \\log_{' + a + '}{(' + t.plusminus() +
                 '+' + y + 'x' + ')}+\\log_{' + a +'}{' + a.pow(c) +
                '}=\\log_{' + a + '}{((' + t.plusminus() + '+' + y + 'x' + ')}\\cdot' +
                 '{' + a.pow(c) +')} \\ \\Rightarrow$$' +
                '$$\\Rightarrow\\ \\begin{cases} {' + b.plusminus() + '+' +
                 z + 'x' + '>0} \\\\ {' + t.plusminus() + '+' + y +
                'x' +'>0}\\\\ {(' + b.plusminus() + '+' + z + 'x' + ') = (' +
                 t.plusminus() + '+' + y + 'x' + ')}\\cdot' + a.pow(c) +
                '\\end{cases}\\Rightarrow \\begin{cases} {' + z + 'x>' +
                (-b.plusminus()) + '} \\\\ {' + y + 'x>' + (-t.plusminus()) +
                '}\\\\{' + b.plusminus() + '+' + z + 'x' + ' = ' +
                 t.plusminus() * a.pow(c) + '+' + y * a.pow(c) + 'x' +
                '}\\end{cases}\\Rightarrow \\begin{cases} x>\\frac{' +
                 (-b.plusminus()) + '}{' + z + '}' +
                '\\\\~\\\\ x>\\frac{' + (-t.plusminus()) + '}{' + y + '}
                \\\\~\\\\ {' + (-y * a.pow(c) + z).plusminus() +
                'x=' + (t * a.pow(c) - b) + '}\\end{cases}\\Rightarrow' +
                '\\begin{cases} x>' + ans + ' ' + ans1 + '\\\\~\\\\
                 x={' + (t * a.pow(c) - b) / ((-y * a.pow(c) + z).plusminus()) +
                '}\\end{cases}$$' +

                'Ответ: $x=' + (t * a.pow(c) - b) /
                ((-y * a.pow(c) + z).plusminus())+'.' + '$',
        });
    });
})();
```

Бригада маляров красит забор длиной 145 метров , ежедневно увеличивая норму покраски на одно и то же число метров. Известно, что за первый и последний день в сумме бригада покрасила 58 метров забора. Определите, сколько дней бригада маляров красила весь забор.

Рабочие прокладывают тоннель длиной 140 метров, ежедневно увеличивая норму прокладки на одно и то же число метров. Известно, что за первый день рабочие проложили 11 метров туннеля. Определите, сколько метров туннеля проложили рабочие в последний день, если вся работа была выполнена за 7 дней.

Саше надо решить 390 задач. Он решает на одно и то же количество задач больше по сравнению с предыдущим днем. Известно, что за первый день Саша решил 6 задач. Определите, сколько задач решил Саша в последний день, если со всеми задачами он справился за 13 дней.

```
class ArithmeticProgression {
    constructor(first, difference) {
        this.first = first;
        this.difference = difference;
        this.nmember = 0;
    }

    member(nmember) {
        this.nmember = nmember;
        return this.first + this.difference * (this.nmember - 1);
    }

    summa(nmember) {
        this.nmember = nmember;
        return ((2 * this.first + this.difference *
         (this.nmember - 1)) / 2) * this.nmember;
    }
}
let a;
let b;
let n = -1;
let str = prompt('Введите первый член прогрессии.', '');
a = Number(str);
str = prompt('Введите разность арифметической прогрессии.', '');
b = Number(str);
while (n <= 0) {
    str = prompt('Введите номер члена прогрессии.
     Число должно быть больше нуля!', '');
    n = Number(str);
}

let progress = new ArithmeticProgression(a, b);

console.log(progress.member(n));
console.log(progress.summa(n));

```

Бизнесмен Плюшкин получил в 2000 году прибыль в размере 1100000 рублей. Каждый следующий год его прибыл увеличивалась на 20% по сравнению с предыдущим годом. Сколько рублей заработал Плюшкин за 2003 год?

Компания «Альфа» начала инвестировать средства в перспективную отрасль в 2001 году, имея капитал в размере 3500 долларов. Каждый год, начиная с 2002 года, она получала прибыль, которая составляла 100% о капитала предыдущего года. А компания «Бета» начала инвестировать средства в другую отрасль в 2007 году имея капитал в размере 4500 долларов, и, начиная с 2008 года, ежегодно получала прибыль, составляющую 200% от капитала предыдущего года. На сколько долларов капитал одной из компаний был больше капитала другой к концу 2009 года, если прибыль из оборота не изымалась?

```
class GeometricProgression {
    constructor(first, factor) {
        this.first = first;
        this.factor = factor;
        this.nmember = 0;
    }
    member(nmember) {
        this.nmember = nmember;
        return this.first * Math.pow(this.factor, this.nmember - 1);
    }
    summa(nmember) {
        this.nmember = nmember;
        if (this.factor < 0) {
            throw new Error("Нельзя найти сумму знакопеременной прогрессии!");
        }
        return (this.first * (Math.pow(this.factor, this.nmember) - 1)) /
         (this.factor - 1);
    }
    infinitySumma() {
        if (this.factor < 0) {
            throw Error;
        }
        if (this.factor > 1) {
            throw new Error("Прогрессия не является бесконечной!");
        }
        return this.first / (1 - this.factor);
    }
}
let a;
let b;
let n = -1;
let str = prompt('Введите первый член прогрессии.', '');
a = Number(str);
str = prompt('Введите множитель прогрессии.', '');
b = Number(str);
while (n <= 0) {
    str = prompt('Введите номер члена прогрессии.
     Число должно быть больше нуля!', '');
    n = Number(str);
}

let progress = new GeometricProgression(a, b);

console.log(progress.member(n));
try {
    console.log(progress.summa(n));
    console.log(progress.infinitySumma());
} catch (error) {
    if (error.name == "Error") {
        console.log(error.message);
    } else {
        throw error;
    }
}
```

Найдите точку максимума функции

$$y = (5\,x - 2)\cos x - 5\sin x + 55,$$

принадлежащую промежутку $(0; \frac{\pi}{2})$.

$$y' = (5\,x - 2)'\cos x + (5\,x - 2)(\cos x)' + (-5\sin x)'$$

$$y' = 5\cos x - (5\,x - 2)\sin x - 5\cos x \Rightarrow\ y' = -(5\,x - 2)\sin x$$

$$y' = 0 \Rightarrow\ (5\,x - 2)\sin x = 0$$

$$1)\quad 5\,x - 2 = 0 \Rightarrow\ 5x = 2 \Rightarrow\ x = 0.4$$

$$2)\quad \sin x = 0 \Rightarrow\ x = \pi n, \quad n \in \mathbb{Z}$$

Промежутку $(0; \frac{\pi}{2})$ принадлежит только точка $0.4$. Подставляя любые значения из промежутка $(0; 0.4)$ в найденную производную, мы узнаём, что она имеет положительный знак, а на промежутке $(0.4; \frac{\pi}{2})$ отрицательный. Производная функции меняет знак с положительного на отрицательный, значит, $0.4$ - это искомая точка максимума.

```
(function() {
    'use strict';
    var arr1 = ['максимума', 'минимума'];
    var arr2 = ['положительный', 'отрицательный'];
    var maxmin1 = sl(0, 1);
    var maxmin2 = (maxmin1 + 1) % 2;
    var ans = sl(2, 14, 2) / 10;
    var b = sl(5, 50, 5);
    var a = ans * b;
    var d = sluchch(-99, 99);
    if (maxmin1 == 1) {
        b = -b;
        a = -a;
    }

    function plusmin(member) {
        var el = math.parse(member);
        el = math.simplify(el);
        el = math.simplify(el, [{
            l: 'n1*n2',
            r: 'n1 n2'
        }]);
        return el.toTex({
            implicit: 'hide'
        });
    }
```

```
    chas2.task.setTask({
        text: 'Найдите точку ' + arr1[maxmin1] + ' функции' +
            '$$ y=(' + plusmin(b + 'x-' + a) + ')' + '\\cos ' +
            plusmin('x-' + b) + '\\sin ' + plusmin('x+' + d) +
            ', $$' +
            'принадлежащую промежутку $(0; \\frac{\\pi}{2} )$.',
        analys: '$$ y^{\'}=(' + plusmin(b + 'x-' + a) +
            ')^{\'}\\cos x+(' + plusmin(b + 'x-' + a) + ')' +
            '(\\cos x)^{\'}+(' + -b + '\\sin x)^{\'}$$' +
            '$$ y^{\'}=' + b + '\\cos x-(' + plusmin(b + 'x-' + a) +
            ')' + '\\sin ' + plusmin('x-' + b) + '\\cos x' +
            ' \\Rightarrow\\ ' + 'y^{\'}=-(' + plusmin(b + 'x-' + a) +
            ')\\sin x$$' + '$$ y^{\'}=0' + ' \\Rightarrow\\ ' + '(' +
            plusmin(b + 'x-' + a) + ')\\sin x=0$$' + '$$1) \\quad' +
            plusmin(b + 'x-' + a) + '=0' + ' \\Rightarrow\\ ' + b +
            'x=' + a + ' \\Rightarrow\\ ' + 'x=' + ans + '$$' +
            '$$2) \\quad \\sin x=0' + ' \\Rightarrow\\ ' +
            'x=\\pi n, \\quad n \\in \\mathbb {Z}' + '$$' +
            'Промежутку $(0;\\frac{\\pi}{2} )$ принадлежит только точка $' +
            ans + '$. Подставляя любые значения из промежутка $(0;' +
            ans + ')$ в найденную производную, мы узнаём, что она имеет ' +
            arr2[maxmin1] + ' знак, а на промежутке $(' + ans +
            '; \\frac{\\pi}{2} )$ ' + arr2[maxmin2] +
            '. Производная функции меняет знак с ' + sklonlxkand(
                arr2[maxmin1]).re + ' на ' + arr2[maxmin2] + ',' +
            ' значит, $' + ans + '$ - это искомая точка ' + arr1[
                maxmin1] + '.',
        answers: ans,
    });
})();
```