

Педагогический проект
на тему:
“Организация IT-турниров по развитию навыков устного счёта на базе
разработанного веб-приложения «ustns4et.ru»”

Побойкин Владимир Яковлевич,
учитель математики,
МБОУ ЗАТО г. Североморск «Лицей №1»

Мурманск
2024
Содержание.

Введение.	2
Актуальность темы проекта	2
Задачи проекта	3
Цель и план разработки проекта	6
Методические проблемы, решаемые проектом	6
Разработка и реализация проекта.	8
1. Написание метода генерации примеров	8
2. Инициализация проекта Django	12
3. Подключение проекта к MySQL, создание моделей	13
4. Размещение проекта на сервере	15
Заключение	16
Список используемой литературы	19
Приложение 1.	20
Приложение 2.	23

Актуальность темы проекта. Ни для кого не является секретом, что в наше время крайне быстро развиваются области IT-технологий, что совсем неплохо. Однако, встает всё большая проблема с сфере образования, связанная с развитием когнитивных функций человека. Практически каждый человек имеет в своем кармане устройство, через которое есть доступ к информации всего мира и умеющее производить невероятные математические расчеты, причем крайне быстро. В виду чего, ленивый мозг человека автоматически отключает ряд, полезных для дальнейшего развития, функций. «Умение запоминать – не нужно, в умении устно считать тоже нет необходимости» - шепчет наше внутреннее, подсознательное, ленивое я.

Работая 18 лет в образовании эта «тенденция» четко прослеживается от начала двухтысячных до нынешнего момента. Когда доступ к интернету был у единиц, студенты, в большинстве своем, могли запоминать по 30-40 ответов на вопросы. Спустя лет 5, когда интернет появился практически в каждом доме, многие студенты еле справлялись с запоминанием 10-15 ответов. Спустя еще некоторое время, когда гаджеты появились у каждого учащегося, даже конкретный билет, с заведомо данными вопросами, стало заучить, или тем более понять, стало невозможным. И, к сожалению, это характерно для большей массы учеников и студентов. Сама система образования загружена выполнением ФГОС (Федерального Государственного Общеобразовательного Стандарта), а не тем, чтобы учить ребят хотя бы простейшему. Значительная часть старшеклассников, на вопрос: «Сколько будет $99 \cdot 7$, скорее всего ответят неверно, поскольку начнут считать, как $9 \cdot 7 \cdot 10 + 9 \cdot 7 = \dots$, напроочь забыв, а для многих, и не зная, что такое умножение. Ведь куда проще из семисот вычесть семь, - это и будет сто семерок без одной, то есть девяносто девять семерок: $99 \cdot 7 = 700 - 7 = 693$. Сколько раз, мой педагогический опыт показывал, насколько важно развивать у ребят знания, умения и навыки быстрого решения простых, базовых задач. Конечно, тогда и более сложные понимаются и решаются куда проще. Только имея крепкий фундамент в основании, мы можем на нем построить что-то грандиозное. В подтверждение этому есть ряд подтверждающих эту теорию

примеров из моего педагогического опыта. Подробнее о них я напишу в заключении.

Возник вопрос, как создать приложение, которое тренировало бы ребят и взрослых устному счету, при этом был стимул учащихся не бросать начатое. Конечно, нужно реализовывать приложение-игру, поскольку игровая, соперническая форма больше стимулирует к участию.

С учетом вышеизложенного, должен быть выполнен ряд **задач**:

- правильная генерация примеров;
- с точки зрения методики преподавания математики, разбить примеры на блоки (этапы).
- объективное начисление баллов;
- организация мини-турниров.

Разберем эти четыре пункта подробнее.

- 1) Большая доля приложений тренировки счёта, которые можно встретить в интернете, имеют крайне случайную генерацию: примеры могут повторяться, или переставленные местами слагаемые уже идут, как различные задания. Несколько подряд идущих примеров могут содержать один и тот же ответ. Сложность заданий скачет произвольно, то есть могут идти простые задания, затем посложней, следом вновь простые и всё это на одном уровне. Становится понятно, что при подобном формировании примеров, объективно оценить кто справляется лучше, кто хуже практически невозможно. Ведь менее сильному пользователю может просто сильно повезти, и его результат окажется лучше. Встает одна из главных целей работы: создать генерацию заданий так, чтобы миновать вышеупомянутые недочеты.
- 2) Примеры лучше разбить на несколько уровней. Причем с увеличиваемой сложностью заданий. Например: разминка, сложение-вычитание, умножение-деление с постепенным увеличением сложности, в конце — уровень для сильных. Нужно отметить, что приложение должно учитывать умение использовать различные способы решений, для этого именно

скорость выполнения заданий должна влиять на итоговый результат. К примеру, большая часть моих учеников за две-три секунды ответят сколько будет $48 \cdot 52$. Да, ведь здесь, всего-навсего нужно воспользоваться формулой разности квадратов, которая изучается в седьмом классе. Приложение должно помочь отработать навыки устного счета так быстро, что вычислять эти примеры, скажем, на калькуляторе было бы куда дольше. Для справки: итак, $(a - b) \cdot (a + b) = a^2 - b^2$, отсюда следует $48 \cdot 52 = (50 - 2) \cdot (50 + 2) = 50^2 - 2^2 = 2500 - 4 = 2496$.

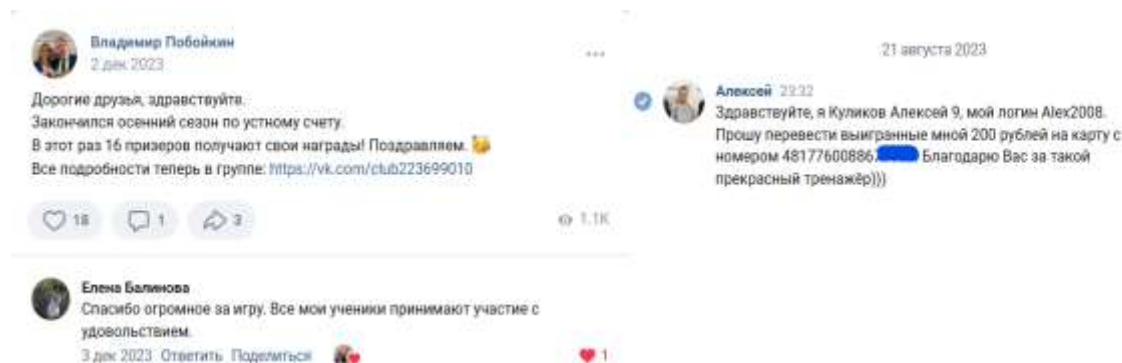
Как показала практика, научиться подобной операции в уме не составляет трудности. Главное хорошо знать компоненты числа, что развивают и закрепляют первые этапы приложения. Самое примечательное, что подобных применений формул просто нет в методике. Однако, ученикам нравится перемножать двузначные числа практически мгновенно, поэтому счел необходимым включить и подобные примеры. Некоторые «нематематики» говорят, мол, это ментальная математика. Нет! Это всего-навсего базовые определения и формулы.

- 3) Начисление очков в приложении напрямую зависит от скорости прохождения этапов. Чем больше практиковаться, тем быстрее проходится тот или иной уровень. Поэтому на каждый этап отводится определенное количество секунд, за которые необходимо решить десять примеров. Оставшееся количество секунд умножается на 1000. Точность оставшегося времени должна измеряться в тысячных долях секунды. За ошибки начисляются штрафы: за первую ошибку – одна секунда задержки (с загоревшимся предупреждением «ошибка»), за вторую – загорается правильный ответ и списывается от оставшегося десятая часть начального времени. Таким образом, специально отвечая неправильно на все задания, даже очень быстро, пользователь не наберет за уровень ни одного балла. В конце игры начисляются суммарные баллы за все уровни, плюс небольшой бонус за безошибочные ответы, то есть за примеры, правильный ответ на

которые был дан с первого раза. Разумеется, возникла необходимость в топовых таблицах, как общей, так и по каждому этапу в отдельности.

- 4) Еще с прошлого учебного года на базе приложения организуются турниры для учащихся, в которых принимают участие не только мои ученики, но и из различных других городов. Сами-собой участники не появились, периодически запускалась реклама в ВК.

Пара отзывов:



В целом, турниры показали себя достаточно хорошим мотиватором. В турнире, который закончился 15.04.24г. Участвовало 160 человек.

Достижения участников в текущем сезоне		
В таблице отображается Ваш лучший результат		
1) Красильникова Полина 10у	675776	60/60
2) Ткачёва Кира 10у	665336	60/57
3) Захаров Егор 10а	658962	60/59
4) Бунькова Мирослава 11А	655619	60/57
5) Штерн Анастасия 10гамма	654910	60/58
6) Колмакова Ксения 10Гамма	647184	60/59
....		
158) Копьев Матвей 8К	32439	18/15
159) Буртасова Ксения 6	28334	25/21
160) Сидорова Варя 5Б	26209	15/12
161) Терентьева Лидия 1В	19368	17/17
162) Аюпов Тимур 6	1900	24/19

С арифметикой всё верно. Десять фейковых участников, появляются в самом начале турнира, с крайне низкими результатами, дабы первый участник не чувствовал крайнего одиночества. Плюс один участник не счет – это Побойкин

Владимир Яковлевич. Резюмируя все изложенное выше, можно сформулировать **цель**: разработать и апробировать тренажер, развивающий такие умения и навыки устного счета, которые не изучаются в образовательном процессе. Реализовать возможность участия в мотивационных турнирах всех зарегистрированных пользователей.

Отталкиваясь от указанных условий-требований, **план** реализации проекта можно изложить в следующих пунктах.

- 1) Написать на python метод генерации заданий для игры, с учетом особенностей приложения.
- 2) Инициализировать проект Django, подключиться к MySQL, написать модели, создать миграции.
- 3) Создать необходимые представления, шаблоны и маршруты
- 4) Настроить панель администратора.
- 5) Написать кода, реализующий процесс игры, на JavaScript.
- 6) Разместить проект на облачном сервере TimeWeb cloud.
- 7) Запустить проект.
- 8) Организовать соревновательные турниры.

Проект написан на Django, построение основано на функциональном представлении. То есть на функциях Python, которые принимают объект запроса и возвращает объект ответа. Модуль игрового процесса использует JS, поскольку требуется быстрая динамика обработки вводимых данных.

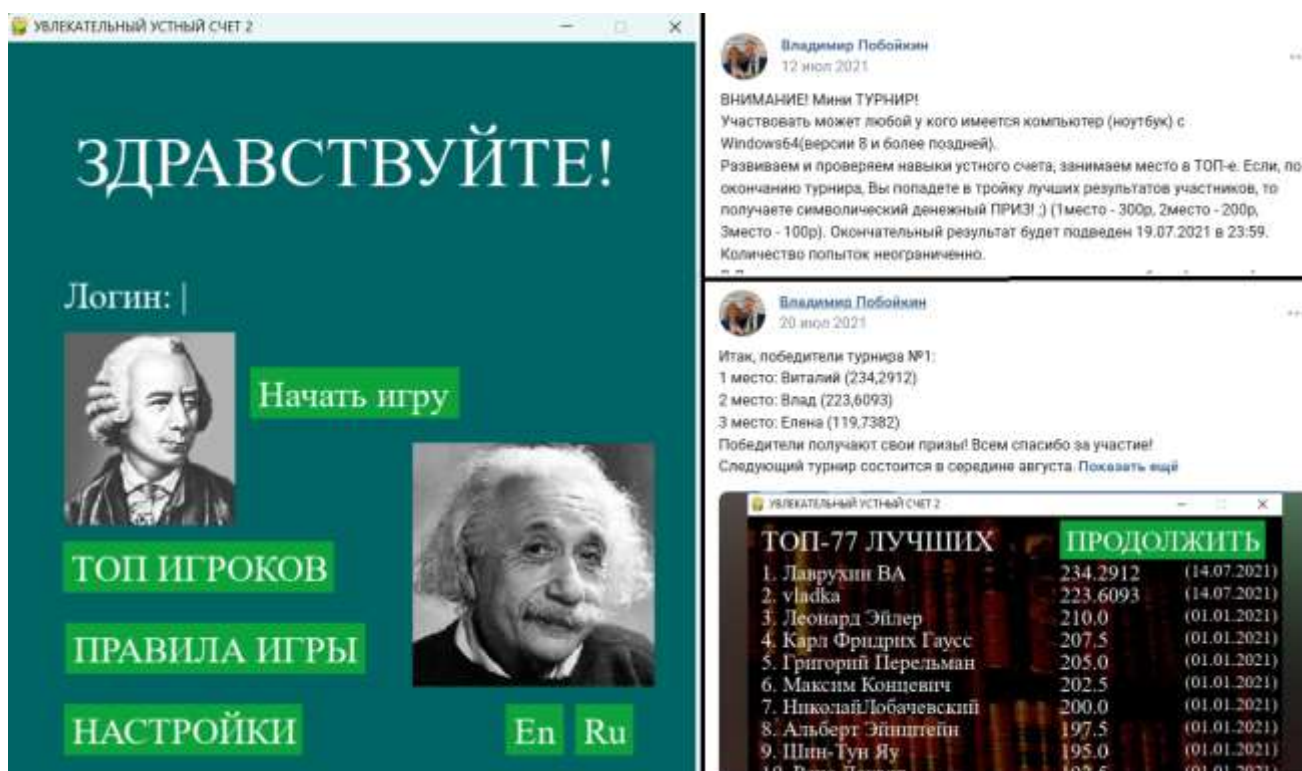
Проблема, которую решает приложение:

Развить у школьников и студентов навыки устного счета, крайне необходимые для успешного и быстрого понимания всех разделов математики, как школьной, так и высшей.

Предпосылки к разработке представляемого проекта, появились еще несколько лет назад:

- 2019 год: ученики решали карточки, написанные вручную в word-e,
- 2020 год: создан простой генератор примеров на python,

- 2021 год: запущено первое приложение, написанное с помощью библиотеки rугате, и конвертированное в исполняемый файл с помощью ауто-ру-to-ехе, организован первый мини-турнир.



Участников было лишь несколько, поскольку, скачивание ехе-файла и его запуск, игнорируя предупреждения windows, оказалось весьма сложной процедурой для пользователей. Следует отметить, что оригинальность (но, не безопасность) этого приложения заключалась в том, что само ехе-приложение работало с MySQL базой расположенной на beget хостинге.

- 2022 год: самостоятельное изучение фреймворка и JavaScript заняло довольно много времени, поэтому первая версия web-приложения (по неизменяемому адресу <https://ustns4et.ru/>) стартует лишь на рубеже 2022, 2023 годов.

Для воплощения проекта в жизнь были использованы фреймворк Django и JS. Изначально проект был размещен на хостинге TimeWeb, затем на облачном сервере TW Cloud. Пользоваться приложением может любой желающий.

QR-код, ссылка на сайт находятся в [приложении 2](#).

Разработка и реализация проекта.

Написание метода генерации примеров

Изначально в приложении планировалось пять уровней, которые должен пройти пользователь, первые три из которых на сложение и вычитание, четвертый на умножение, пятый на деление. Основная задача этого вопроса состояла в том, что все примеры должны быть различными, и совокупный уровень сложности соответствующих этапов должен быть примерно одинаковым. Для генерации заданий служит класс Examples. Что не вписывается в концепцию функционального представления, однако на работу приложения это никак не влияет.

1 уровень: «Разминка». Десять примеров (1 – 10), из которых пять на сложение и пять на вычитание. Здесь, все операции проходят в рамках первого десятка.

```
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225

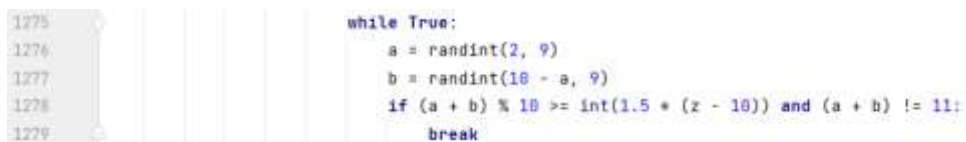
if m == 0:
    provsov = [[0] * 3 for i in range(11)] # обнуление массива контроля повторений
    prcoin = [0] * 10 # каждая цифра попадается не более 2 раз
    for z in range(0, 5):
        while True:
            fl = 0
            ff = 0
            while (ff == 0):
                a = randint(2, 8)
                b = randint(1, 10 - a)
                if prcoin[a] < 3 and prcoin[b] < 3:
                    prcoin[a] += 1
                    prcoin[b] += 1
                    ff = 1
            mas = provsov
            d = 0
            while (d < 5): # все 5 используем предыдущие
                if (mas[d][0] == a and mas[d][1] == b) or (mas[d][0] == b and mas[d][1] == a) \
                    or (mas[d][2] == a + b):
                    prcoin[a] -= 1
                    prcoin[b] -= 1
                    fl = 1
                    d = 10
                d += 1
            if fl == 0:
                provsov[z][0] = a
                provsov[z][1] = b
                provsov[z][2] = a + b
                tasks[z][r] = str(a) + '+' + str(b) + '='
                otvs[z][r] = '0' # otvs[y + z][r] = str(a + b)
            break
```

Здесь два списка *provsov* и *prcoin* контролирую уникальность при генерации заданий: *provsov* – не допускает совпадений в примерах, причем учитывается переместительное свойство сложения, а также ответы на все пять первых

примеров – попарно различны (строки 1212 - 1213). Последнее указанное условие не допускает одинаковых ответов в подряд идущих примерах, что может привести к случайному, более быстрому прохождению уровня. Следует отметить, что этот принцип не повторяющихся ответов распространяется на все этапы приложения. *prosoip* не допускает повторения одного числа более чем два раза. По скорости выполнения данный способ формирования заданий является довольно медленным, т.к. выполнения условий проверяются после очередного заполнения случайными значениями. Однако он выполняет лучшую вариативность и не влияет на видимую производительность.

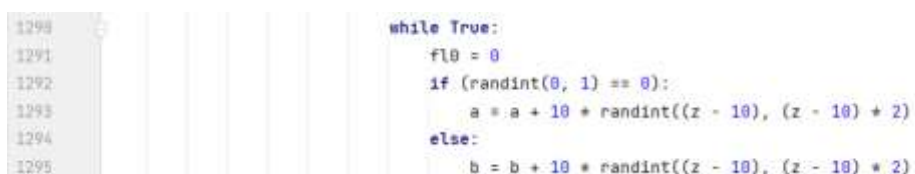
Аналогично, в строках с 1227 по 1257 генерируются примеры на вычитание.

2 уровень: «Сквозь десятки». Здесь также десять примеров (11- 20), пять из которых на сложение и пять на вычитание. Как говорит название уровня, сами примеры формируются так, что ответ должен получаться с увеличением или уменьшением десятка. Одно число в условии однозначное, другое двузначное. Первым делом задаются разряды единиц, причем с незначительным увеличением последней цифры результата (строка 1278).



```
1275 while True:
1276     a = randint(2, 9)
1277     b = randint(10 - a, 9)
1278     if (a + b) % 10 >= int(1.5 * (z - 10)) and (a + b) != 11:
1279         break
```

Также в этой строке исключаются числа, оканчивающиеся на единицу. С точки зрения методики преподавания математики, такие числа даются проще. Стараюсь минимизировать возможность того, что кому-то повезет больше, нежели другим. Далее, случайным образом, для одного из чисел добавляются десятки.



```
1298 while True:
1299     fl0 = 0
1300     if (randint(0, 1) == 0):
1301         a = a + 10 * randint((z - 10), (z - 10) + 2)
1302     else:
1303         b = b + 10 * randint((z - 10), (z - 10) + 2)
```

Аналогично, в строках с 1319 по 1374 создаются задания подразумевающие вычитания через десятков (примеры с 15 – 20).

3 уровень. «Минуя сотни». Опять же десять примеров (21 – 30), пять из которых на сложение и пять на вычитание. Только здесь генерируются трехзначное и одно-двузначное число (однозначное только в двух случаях: 21 и 26 примеры).

```

1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390

while True:
    fl = 0
    while True:
        a = randint(5, 9)
        b = randint(10 - a, 9)
        if (a + b) % 10 >= int(1.5 * (z - 20)) and ((a + b) % 10 != 0):
            break
    i = randint(int((z - 18) * 0.7), z - 19) * 100
    a = 100 - 18 * (randint(int((z - 20) * 0.35), int((z - 20) * 0.6)) + 1) + a
    b = randint(int((z - 19) * 0.5), int((z - 19) * 0.9)) * 10 + b
    prst = str(a + i - 100 + b)

```

Здесь в строках 1382 и 1383 задаются значения в разрядах единиц, в строке 1386 переменной i присваивается случайное количество сотен в трехзначном числе. В строках 1387 и 1388 определяется количество десятков в числах. Переменная z имеет значение номера примера в игре. Все коэффициенты: 0.7, 0.35, 0.6, 0.5, 0.9 на которые умножается очередное значение $(z - a)$, подобраны таким образом, чтобы границы случайного диапазона расширялись и несколько сдвигались вправо левая граница, что увеличивает сложность примеров с приближением таковых к концу этапа. Следует отметить, приведенный алгоритм подразумевает генерацию чисел таким образом, что для получения правильного ответа необходимо увеличить на 1 и разряд десятков, и разряд сотен. Например: $387 + 48 = ?$ Каждый пример формирует лучшее представление об арифметических операциях и составе числа: $387 + 48 = (387 + 13) + (48 - 13) = 400 + 35 = 435$.

Пять примеров (25 – 30) на вычитание, в строках с 1409 по 1435 генерируются аналогичным алгоритмом, единственное, что результатом теперь является двузначное число, полученное из разности трехзначных:

```

1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499

tasks[z][r] = str(a + b) + '-' + str(a) + '='
otvs[z][r] = str(b)

```

4 уровень: «Хитрое умножение». Десять примеров (31 – 40), пять из которых имеют вид $a \cdot b$, где $a = (n \cdot 100 + k)$, $n \in \{1, 2, 3, 4, 5\}$, $k \in N$, $k \in [1, 9]$, $b \in N$, $b \in [2, 9]$ следующие пять: $a \cdot b$, где $a = (n \cdot 100 - k)$, $n \in \{1, 2, 3, 4, 5\}$, $k \in N$, $k \in [1, 4]$, $b \in N$, $b \in [2, 9]$. Реализация приведена в строках с 1436 по 1485

```

1441
1442
1443
1444
fl = 0
a = randint(1, 9)
a = 100 + a + 100 * randint(int((z - 30) * 0.4), int((z - 30) * 0.8))
b = randint(2, 9)

```

Строка 1443 повышает количество сотен в примерах «плавным» образом. Здесь участнику необходимо вспомнить, что такое умножение, и когда легче вычитать, нежели умножать в столбик. Например: $98 \cdot 7 = 700 - 14 = 686$.

5 уровень: «"Life hack"-Деление». Вновь игрока ожидают десять примеров (41 – 50), пять из которых имеют вид a/b , где $a = (n \cdot 100 + k)$, $n \in \{1, 2, 3, 4, 5\}$, $k \in \mathbb{N}$, $k \in [1, 9]$, $b \in \mathbb{N}$, $b \in [2, 9]$ последующие пять: a/b , где $a = (n \cdot 100 - k)$, $n \in \{1, 2, 3, 4, 5\}$, $k \in \mathbb{N}$, $k \in [1, 4]$, $b \in \mathbb{N}$, $b \in [2, 9]$. Реализация приведена в строках с 1487 по 1530

```

1493
1494
1495
a = randint(1, 9)
a = 100 + a + 100 * randint(int((z - 40) * 0.4), int((z - 40) * 0.8))
b = randint(2, 9)

```

Также, как и в четвертом уровне, сдвигается левая граница и расширяется сам диапазон, входящий в функция `randint`, которая отвечает за число сотен (строка 1494). Таким образом, чем дальше пример от начала уровня, тем больше делимое.

Необходимо отметить, что со второго по пятый уровни реализуется проверка вида:

```

prst = str(a * b)
for d in range(0, 10):
    if prst.find(str(d) + str(d)) != -1: fl = -1

```

Переменная *prst* содержит строковое значение ответа, которое должен ввести пользователь. Именно эта часть кода отвечает за то, что ответ не должен содержать двух подряд идущих одинаковых цифр. Это, опять же, сделано для того, чтобы свести практически к нулю вероятность появления простых примеров. Поскольку, как показывает практика, с примерами, имеющими в ответе подряд идущие одинаковые цифры, игроки справляются быстрее.

6 уровень появился уже по прошествии нескольких первых сезонов. Он также представляет десять примеров (51 – 60), которые требуют некоторых углубленных пониманий арифметики, а в нескольких примерах знание и умение быстро применять некоторых формул средней школы. Полный скрин кода приведен в [приложении 2](#).

Инициализация проекта Django

Необходимо отметить, что следующие три пункта документа описаны для тех, кто уже или планирует развиваться в IT-разработке.

Инициализация проекта, как и большая часть работы над ним, проводилась в среде IDE PyCharm 2021.3.3 (Community Edition). Создание первого, тогда казалось, что пробного проекта, проводилась достаточно давно. В результате «пробный» проект, вырос до вполне действующего (использовались, в миниатюре, AJAX принципы внедрения, т.е. динамические).

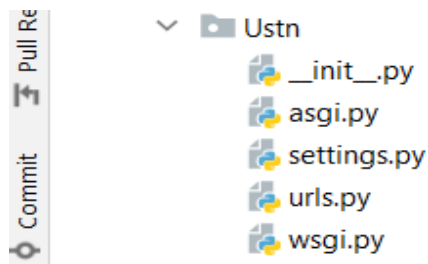
При инициализации были проведены следующие действия:

- установка Django:

pip install Django

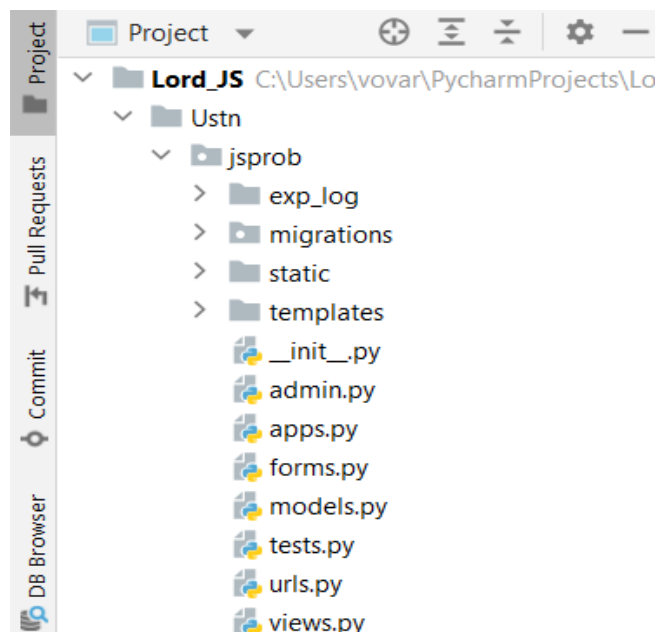
- создание проекта:

django-admin startproject Ustn



- создание приложения jsprob:

python manage.py startapp jsprob



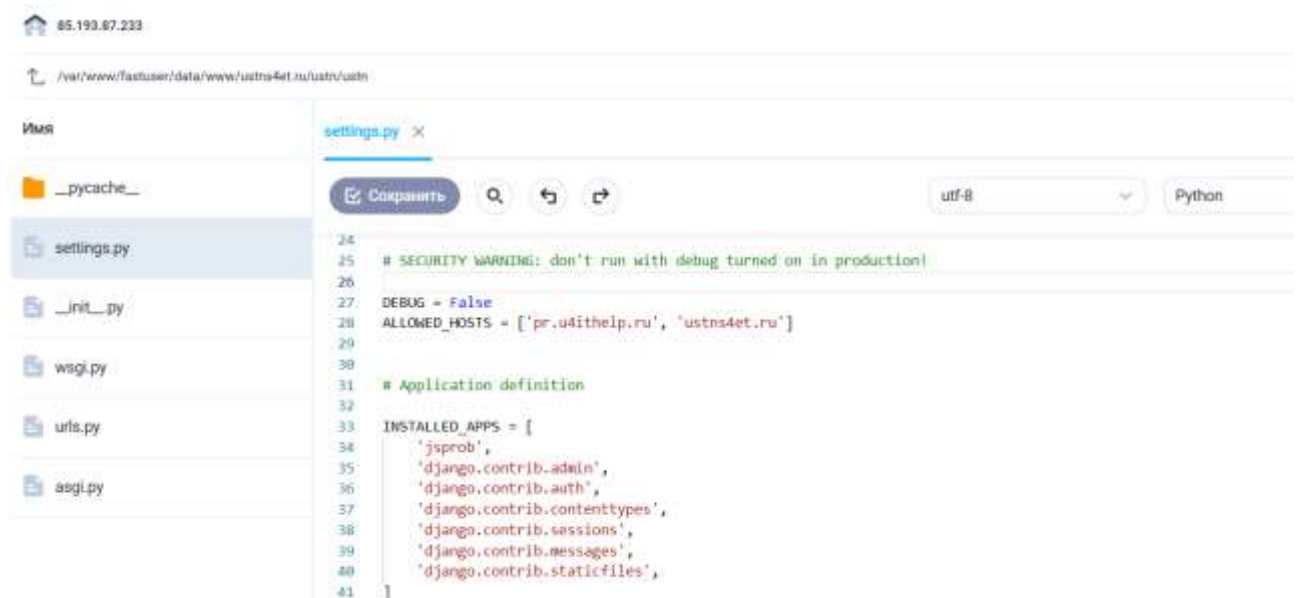
- добавление приложения в setting.py

```

25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = ['*']
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'jsprob',
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41 ]

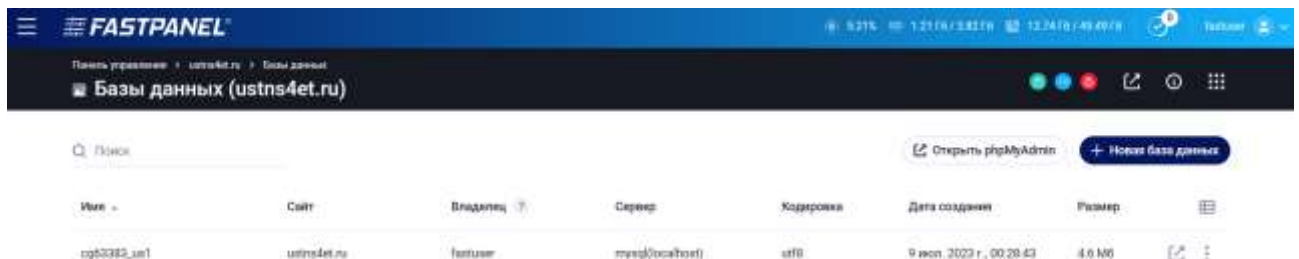
```

Необходимо отметить, что на боевом сервере debug отключен и допустимые хосты ограничены.



Подключение проекта к MySQL, создание моделей.

Приложение использует базу данных на TW Cloud, где и размещен сам проект:



Версия 8.0.36

The screenshot shows the phpMyAdmin web interface. At the top, a green status bar indicates 'Отображено: 1 строк (1 - 0 из всего. Запрос занял 0.0001 сек.)'. Below it, the SQL query 'SELECT VERSION();' is entered. The interface includes tabs for 'Профилирование', 'Планировщик реляционных', 'Изменить', 'Анализ SQL запроса', 'Создать PHP-код', and 'Обновить'. A dropdown menu shows 'Показать количество строк: 25'. The 'VERSION()' query result is displayed as '8.0.36-0ubuntu623041'. Below this, the 'DATABASES' tab is active, showing a tree view of databases on the left and a table of database information on the right. The table lists databases like 'auth_group', 'auth_group_permissions', 'auth_permission', 'auth_user', 'auth_user_permissions', 'django_admin_log', 'django_content_type', 'django_migrations', 'django_session', 'information_schema', and 'performance_schema'.

Настройки setting.py

```
74 # Database
75 # https://docs.djangoproject.com/en/3.2/ref/settings/#databases
76
77
78 DATABASES = {
79     'default': {
80         'ENGINE': 'django.db.backends.mysql',
81         'NAME': 'cg63383_uslh',
82         'USER': 'cg63383_uslh',
83         'PASSWORD': 'LordOnlyus4',
84         'HOST': '5.23.50.121',
85         'PORT': '3306',
86         'OPTIONS': {
87             'init_command': "SET sql_mode='STRICT_TRANS_TABLES'",
88         }
89     }
90 }
```

Замечание для разработчиков. Здесь, на отладочном сервере, указана тестовая БД. На боевом указана активная БД и, конечно 'HOST': 'localhost'.

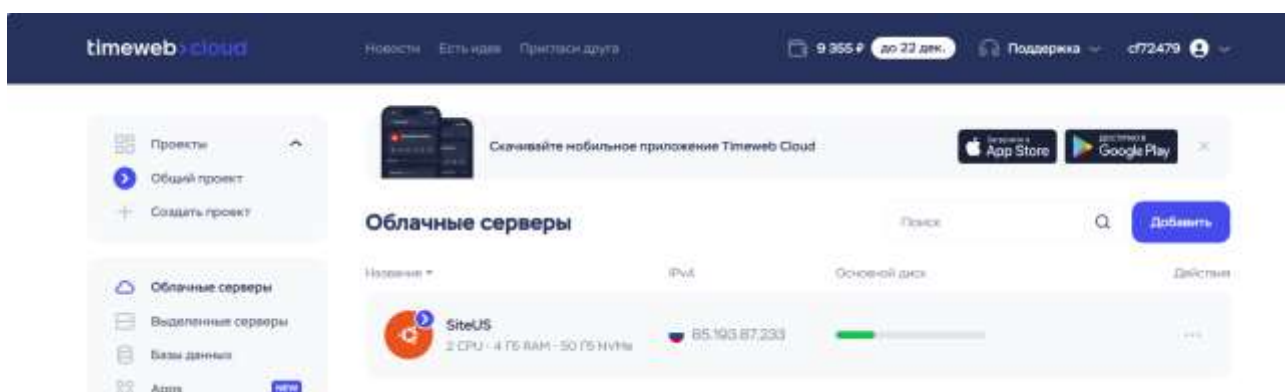
Размещение проекта на севере.

Изначально проект был размещен на хостинге TimeWeb



В точности по видео-инструкции: [Как разместить готовый сайт на Django на хостинге - поиск Яндекса по видео \(yandex.ru\)](#)

В данный момент проект перенесен на облачный сервер TW Cloud



Операционная система Ubuntu 22.04.2 LTS + FASTPANEL.

Заключение:

Приложение, которому посвящен данный проект, смогло достаточно в полной мере реализовать основные [цели](#). Участвовать в турнирах и усовершенствовать свои навыки устного счета может каждый желающий:

Адрес сайта: [ТУРНИР ТРЕНАЖЕР ПО УСТНОМУ СЧЕТУ \(ustns4et.ru\)](https://ustns4et.ru).



Более ранние воплощения проекта в жизнь не имели успеха, поскольку это был конвертированный в exe-файл код python. Его и сейчас можно скачать по ссылке: <https://disk.yandex.ru/d/B8BjlizaOdgwog> (проигнорируйте предупреждения).

Изучение фреймворка Django и основ JS (прошел 1,5 годовую профессиональную переподготовку), позволило создать веб-приложение, доступ к которому, осуществляется практически мгновенно для любого заинтересовавшегося человека. Поскольку основной смысл приложения заключается в быстром решении примеров, реализация точного таймера через браузер казалась сомнительной, однако совместное использование системного таймера устройства и внутреннего счетчика JS дало положительный результат. Django позволил реализовать пользовательский интерфейс и быстрое взаимодействие с MySQL. Для приложения был приобретен домен ustns4et.ru (“ustn” – устный, “s4et” – счет).

Приложение адаптировано, как стационарным компьютерам и ноутбукам, так и различного рода гаджетам.

Первая веб-версия впервые была запущена в прошлом учебном году, и несколько небольших турниров было проведено еще до лета прошлого 2023 года. Победительница этих турниров Яна К, ученица выпускного класса, (ее результат

можно найти в глобальном топе приложения), написала ЕГЭ с лучшим результатом в муниципалитете. И, напротив, школьники, имеющий крайне низкий результат, и не желающие его исправить, то есть не тренирующиеся, так и остаются с крайне низким результатом по всем разделам математики. На данный момент пальму первенства держит Полина К из десятого класса. Как призналась сама Полина, когда прошла на областную олимпиаду, куда ранее никогда не попадала: «Это все тренажер устного счета». Отрыв от соперников довольно значительный. Текущий лидер шла к этому результату постепенно и усердно. На 04.10.2024 Полина запускала игру 20297 раз, прошла целиком 1192 раза. Следует отметить, что на ЕГЭ 2025 года Полина набрала 88 баллов (профиль был сложным) – это лучший результат оп Североморскому муниципалитету.

Таким образом, проект подтвердил еще две негласные гипотезы:

– у современных школьников (и не только) отсутствует базовый фундамент математики, а именно, не отработаны до автоматизма арифметические операции. И это на самом деле так, старшеклассники лицея, где Ваш покорный слуга проводил открытый урок, могут только в столбик вычислять примеры вида:

$99 \cdot 7 =$ или $314 - 278 =$. Укрепление же этого «фундамента», то есть отработка умений и навыков в области простейшей математики, дает возможности успешно решать и сложные задачи, а также усваивать новый материал куда быстрее нежели ранее.

– в юном возрасте, обучение происходит быстрее и с куда лучшим эффектом. Ни один педагог старше тридцати пяти лет, как бы ни тренировался, не может показать результат выше 620000.

Аутентификация на сайте осуществляется лишь при вводе уникального логина, который знает только пользователь. Это сделано для упрощения входа в аккаунт. Даже, если кто-нибудь подберет логин другого пользователя, то никаким образом навредить он не сможет. Приложение учитывает только лучшие результаты.

Вопрос с нечестными пользователями решается автоматически. Какими мы бы вспомогательными средствами не пользовались, пытались же это делать многие. Это и калькуляторы, и Photomath, даже «Алиса». Больше 500000 набрать не удастся. Поскольку приложение рассчитано на то, что, пройдя тренажер достаточное количество раз, пользователь научиться вычислять подобные примеры в уме, куда быстрее, чем набрать ряд необходимых цифр на калькуляторе и т.п.

На сайте регулярно проводятся турниры, достигая в которых определенных результатов, участники получают символические вознаграждения. Оцениваемые показатели – это не только лучшие результаты по отдельным этапам и игре в целом, но и ежедневное постоянство в прохождении игры. За участие, каждый день суммируются накопительные баллы, которые, в конце сезона переводятся в некоторый приз.

В приложении ненавязчивым образом используется стимуляция к духовно-нравственному росту. Поскольку сам являюсь христианином, служу в Храме, конечно, приложение не обошлось без цитат из Библии, ограничился притчами Соломона. Таковых в базе данных приложения 192, и один из методов периодически, случайным образом, возвращает одну из притч на обозрение пользователю. Это происходит, либо по завершении игры, либо после пройденного этапа, если игрок улучшил свои предыдущие достижения, а также, с вероятностью 1/3, притча появляется на стартовой странице приложения.

На 11.09.2025 сайт насчитывает [более 850 участников](#).

В декабре 2023 года была создана группа ВК, посвященная проекту. В которой любой желающий может ознакомиться с положениями проводимых турниров, а также с результатами проведенных.

Ссылка на группу: [ТУРНИРЫ ПО УСТНОМУ СЧЕТУ \(vk.com\)](#)

На основе рассмотренного проекта реализуется его значительное расширение: [на новом сайте](#) любой учитель сможет организовывать отдельные турниры в своём классе или учебном заведении. Проект будет учитывать разные возрастные уровни участников. Запуск нового сайта запланирован на сентябрь 2025 года.

Список используемой литературы

Книга одностомная:

1. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — Мэтиз Эрик.
2. Django 3.0. Практика создания веб-сайтов на Python. — СПб.: БХВ-Петербург, 2021. — Владимир Дронов.
3. Изучаем программирование на JavaScript. — СПб.: Питер, 2015. — Робсон Элизабет, Фримен Эрик.

Электронные ресурсы:

1. <https://lyceum.yandex.ru/>
2. <https://tvorcheskie-proekty.ru/node/4298>
3. <https://multiurok.ru/files/issledovatelskaia-rabota-po-informatike-sozdanie-w.html>
4. <https://www.youtube.com/watch?v=LFyeHQwo4U&list=PLDyJYA6aTY1nZ9fSGcsK4wqeu-xaJksQQ>
5. <https://docs.djangoproject.com/en/4.1/>
6. <https://habr.com/ru/articles/747234/>

Приложение 1.

Фрагмент кода генерирующего задания для шестого уровня.

```

1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554

if nn == 50:
    # 50 разность квадратов
    a = randint(112, 149)
    tasks[50][r] = str(a) + '^2-' + str(a - 100) + '^2'
    otvs[50][r] = str(((a - 100) * 2 + 100) + 100)

    # 51-52 разность квадратов через симметричные множители
    a = randint(2, 9) * 10
    b = randint(1, 3)
    while True:
        c = randint(2, 9) * 10
        if a != c: break
    d = randint(4, 7)
    if randint(0, 1) == 0:
        tasks[51][r] = str(a - b) + '*' + str(a + b)
    else:
        tasks[51][r] = str(a + b) + '*' + str(a - b)
    otvs[51][r] = str(a * a - b * b)
    if randint(0, 1) == 0:
        tasks[52][r] = str(c - d) + '*' + str(c + d)
    else:
        tasks[52][r] = str(c + d) + '*' + str(c - d)
    otvs[52][r] = str(c * c - d * d)

1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579

# 53-55 6 разностных
s = [0 for i in range(3)]
r1 = [0 for i in range(3)]
while True:
    for d in range(3):
        q = gene(d + 2)
        r1[d] = q[0]
        s[d] = q[1]
    if sum([p / abs(p) for p in s if p > 0]) == 1 \
        and abs(s[0]) != abs(s[1]) and abs(s[0]) != abs(s[2]) and abs(s[1]) != abs(s[2]):
        break
st = ['' for i in range(3)]
for n in range(3):
    for m in r1[n]:
        if m > 0 and m != r1[n][0]:
            st[n] = st[n] + '+' + str(m)
        else:
            st[n] = st[n] + str(m)
tasks[53][r] = st[0]
otvs[53][r] = str(s[0])
tasks[54][r] = st[1]
otvs[54][r] = str(s[1])
tasks[55][r] = st[2]
otvs[55][r] = str(s[2])

1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594

# 56-57 (125 25)
i = randint(1, 4) # {1-125, 125 2-25, 25 3-125, 25(8) 4-125, 25(4)}
if i == 1 or i == 2:
    while True:
        a = randint(2, 24)
        b = randint(2, 24)
        if a != b: break
    while True:
        a1 = randint(2, 12)
        b1 = randint(2, 12)
        if a1 != b1: break
    d = [[a, b], [a1, b1]]
    e = [4, 8]
    c = 25 + (i - 1) * 100

```

```

1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616

    if randint(0, 1) == 0:
        tasks[56][r] = str(c) + '*' + str(400 + d[1 - 1][0] * e[1 - 1])
    else:
        tasks[56][r] = str(400 + d[1 - 1][0] * e[1 - 1]) + '*' + str(c)
    otvs[56][r] = str((400 + d[1 - 1][0] * e[1 - 1]) * c)

    if randint(0, 1) == 0:
        tasks[57][r] = str(c) + '*' + str(800 + d[1 - 1][1] * e[1 - 1])
    else:
        tasks[57][r] = str(800 + d[1 - 1][1] * e[1 - 1]) + '*' + str(c)
    otvs[57][r] = str((800 + d[1 - 1][1] * e[1 - 1]) * c)

if i == 3:
    while True:
        a = randint(2, 24)
        b = randint(2, 12)
        if a != 2 * b and a != b:
            break
    for k in range(56, 58):
        d = [a, b]
        e = [4, 8]
        c = (57 - k) * 100 + 25

        if randint(0, 1) == 0:
            tasks[k][r] = str(c) + '*' + str(800 + d[57 - k] * e[57 - k])
        else:
            tasks[k][r] = str(800 + d[57 - k] * e[57 - k]) + '*' + str(c)
        otvs[k][r] = str((800 + d[57 - k] * e[57 - k]) * c)

if i == 4:
    while True:
        a = randint(2, 24)
        b = randint(2, 12)
        if a != 2 * b and a != b:
            break
    for k in range(56, 58):
        d = [a, b]
        e = [4, 8]
        c = (57 - k) * 100 + 25
        if randint(0, 1) == 0:
            tasks[k][r] = str(c) + '*' + str(400 + d[57 - k] * e[57 - k])
        else:
            tasks[k][r] = str(400 + d[57 - k] * e[57 - k]) + '*' + str(c)
        otvs[k][r] = str((400 + d[57 - k] * e[57 - k]) * c)

# 58 (pachpeñenur99)
while True:
    a = randint(1, 8) * 10 + randint(1, 8)
    b = 99 - a
    c = randint(1, 8) * 10 + randint(1, 9)
    if a != b and a != c and b != c: break
if randint(0, 1) == 0:
    f = str(a) + '*' + str(c)
else:
    f = str(c) + '*' + str(a)
if randint(0, 1) == 0:
    g = str(b) + '*' + str(c)
else:
    g = str(c) + '*' + str(b)
tasks[58][r] = f + '+' + g
otvs[58][r] = str(a * c + b * c)

```

```

1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679

```

```

# 59 (распределител)
while True:
    a1 = randint(1, 8) * 10 + randint(2, 9)
    b1 = 101 - a1
    c1 = randint(1, 8) * 10 + randint(1, 9)
    if a1 != b1 and a1 != c1 and b1 != c1 and a1 != a and c1 != c: break
if randint(0, 1) == 0:
    f = str(a1) + '*' + str(c1)
else:
    f = str(c1) + '*' + str(a1)
if randint(0, 1) == 0:
    g = str(b1) + '*' + str(c1)
else:
    g = str(c1) + '*' + str(b1)
tasks[59][r] = f + '*' + g
otvs[59][r] = str(a1 * c1 + b1 * c1)

for i in range(1, randint(20, 25)):
    while True:
        k1 = randint(50, 59)
        k2 = randint(50, 59)
        if k1 != k2: break
    tasks[k1][r], tasks[k2][r] = tasks[k2][r], tasks[k1][r]
    otvs[k1][r], otvs[k2][r] = otvs[k2][r], otvs[k1][r]

```

Приложение 2.

Ссылка и QR-код на приложение.

Турнир-тренажер устного счета:

<https://ustns4et.ru/>

