

ТЗ на разработку комплекса (базы данных и программной среды) ДЛЯ предиктивной аналитики шахматных партий.

0. Задача в общем подразумевает 2 "блока" формирование самой базы данных, и некоторый "сервис", который будет на входе принимать алгоритмы, взаимодействовать с базой, и на выходе предоставлять результаты.

1. Создать базу, и обновлять данные в режиме он-лайн (шахматные партии) с источника: <https://lichess.org/api>

2. Сделать API/интерфейс/метод взаимодействия с пользователями/другими разработчиками, которые будут предоставлять алгоритмы оценки имеющейся в базе информации.

3. Проводить оценку результатов каждого алгоритма, на основе фактического итога партии. Поскольку на один алгоритм "в вакууме" - это оценить невозможно, оценка будет проводиться методом сравнения алгоритмов между собой (соотв. если их будет 10 или 100 - сравнения каждого с каждым). Пример: [VerWhite, VerDraw, VerBlack], Алгоритм1 [33%, 34%, 33%] , Алгоритм2 [50%, 5%, 45%]. 1. Сравниваем Алгоритм1 с Алгоритмом2: if VerWhite1>VerWhite2 then Value_Алгоритм1_White= (1/VerWhite2) * (результат партии) - 1 else Value_Алгоритм1_White=0 , if VerDraw1>VerDraw2 then Value_Алгоритм1_Draw = (1/VerDraw2) * (результат партии) - 1 else Value_Алгоритм1_Draw=0, if VerBlack1>VerBlack2 then Value_Алгоритм1_Black= (1/VerBlack2) * (результат партии) - 1 else Value_Алгоритм1_Black=0. (результат партии) = 1 для сыгравшего исхода, 0 для остальных исходов. Value_Алгоритм1_Full = Value_Алгоритм1_White + Value_Алгоритм1_Draw + Value_Алгоритм1_Black В данном примере, допустим партия закончилась 1-0 (победой белых). Получим: Value_Алгоритм1_Full = 0 + (-1) + 0 = -1 (в данном случае значения Алгоритма1 относительно к Алгоритму2). Value_Алгоритм1_ToАлгоритм2 как-то так. 2. Аналогично сравниваем и отношение Алгоритма2 к Алгоритму1. Value_Алгоритм2_ToАлгоритм1. if VerWhite2>VerWhite1 ... и т.д. И далее эти значение суммируются и делятся на кол-во партий.

4. Программный комплекс будет определять вероятности шахматных партий. на начальном этапе - только победа_белых/ничья/победа_черных, далее усложним задачи (количество ходов, количество ходов конём ... и т. д.)

5. Должна быть возможность выбора партий (игр) для анализа. Т.е. Не всё подряд (что есть на сайте <https://lichess.org/api>), а выбор — например «анализировать все партии игрока Карлсен», «анализировать все партии такого-то турнира», «анализировать все партии которые сейчас идут в Стриме» и т. п.

6. Работать это должно на выделенном сервере, либо в облаке типа aws.

7. Предоставить код и сопровождающую документацию.

Разделим задачи условно на "базовые" и "опциональные" :

- "базовые" - то что необходимо будет изначально для создания прототипа и тестирования
- "опциональные" - это полный набор желаемых функций которые могут потребоваться (не сейчас)

Что желательно видеть в "базовом" варианте :

- <https://lichess.org/games> (текущие игры)

из них нужны будут :

- <https://lichess.org/games/best>
- <https://lichess.org/games/bullet>
- <https://lichess.org/games/blitz>
- <https://lichess.org/games/rapid>
- <https://lichess.org/games/classical>

Это "базовый" набор игр, которые будем использовать.

(все ссылки - по веб-сайту lichess.org, для определения что нужно будет, через API наверное как-то по другому будут все эти запросы)

Далее: какая информация будет необходима - вся доступная по каждому игроку:

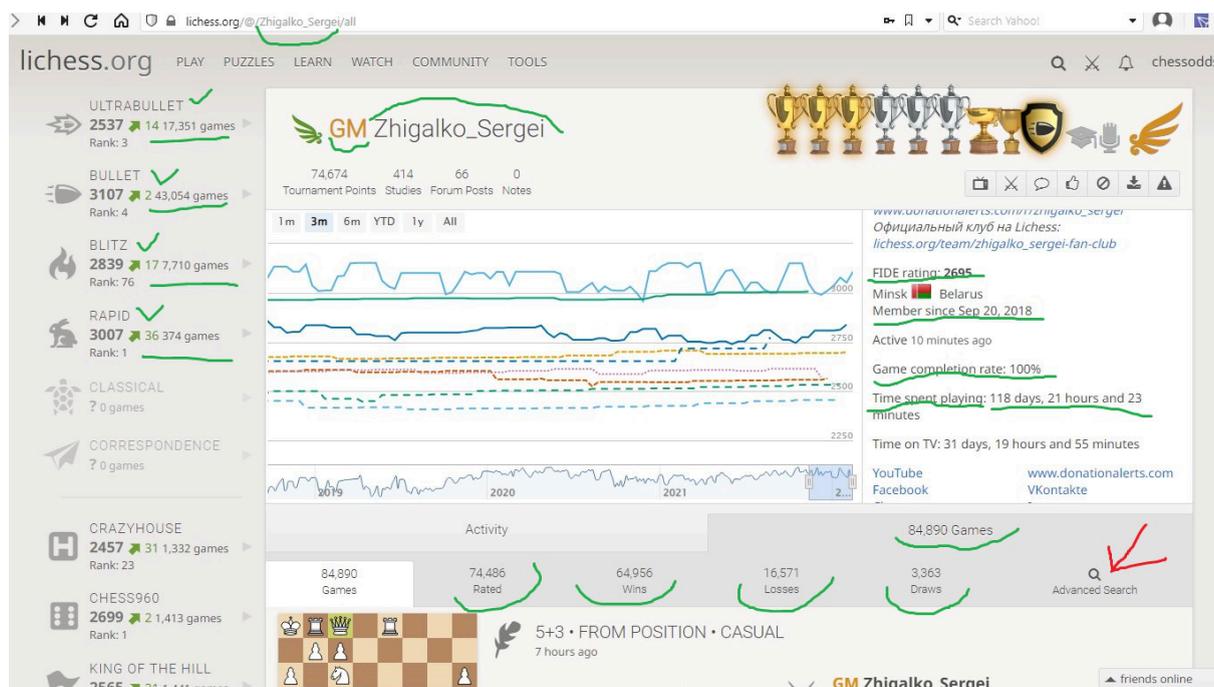
- Имя (ник) игрока
- Титул (если есть GM / IM / и т. п.)
- Рейтинг FIDE (если есть) (уточнение Рейтинг FIDE не является рейтингом lichess, никак не зависит и не влияет одно на другое, нужно

будет исключительно для информационных целей)

- Дата регистрации (member since)
- % Завершённых игр
- Проведённое за игрой время
- Количество сыгранных игр (из них выиграно/ничья/проиграно) (общее)
- Аналогично по каждому типу игр (слева колонки ULTRABULLET, BULLET, BLITZ, RAPID, CLASSICAL)
- Rating (lichess), по каждому типу игр свой рейтинг.
- Rating Progression
- Rating deviation
- Полная история и статистика игр (кнопка **Advanced Search** — все игры, либо **View the games** - уже поделено по типам игр (см. картинки)

Пример:

https://lichess.org/@/Zhigalko_Sergei/all



https://lichess.org/@/Zhigalko_Sergei/perf/bullet

Zhigalko_Sergei Bullet stats

[VIEW THE GAMES](#)

Rating: **3107.15**. Zhigalko_Sergei is better than **100.0%** of Bullet players.

Progression over the last 12 games: **▲2**. Rating deviation: **45.12**.

Total games	46325	Average opponent	2700.03
Rated games	43055 93%	Victories	34290 7▲ friends online

Rating: **3107.15**. Zhigalko_Sergei is better than **100.0%** of Bullet players.

Progression over the last 12 games: **▲2**. Rating deviation: **45.12**.

Total games	46325	Average opponent	2700.03
Rated games	43055 93%	Victories	34290 74% ✓
Tournament games	14357 31%	Draws	2103 5% ✓
Berserked games	2802 20%	Defeats	9932 21% ✓
Time spent playing	39 days, 15 hours and 17 minutes	Disconnections	0 0%

Highest rating: **3181** Lowest rating: **2210**

Информация из истории игр:

- Время игры (1+0, 1/2+0, и т.п)
- Тип игры (Blitz, Bullet и т.п)
- Рейтинговая или нет (RATED / CASUAL)
- Дата игры
- Цвет игры (белыми либо чёрными)
- Rating на момент игры
- Соперник (ник, Rating на момент игры)
- Турнирная игра или нет, в каком турнире игралась (если турнирная)
- Результат игры
- Метод победы (Black time out, White resigned, Checkmate и т.п.)

Пример:

The screenshot shows a search results page on Lichess.org for the user Zhigalko_Sergei. It displays four chess game entries, each with a board diagram, game type, time control, and player information.

- Game 1:** B01 Scandinavian Defense: Mieses-Kotroc Variation. 1. e4 d5 2. exd5 Qxd5 3. Nc3 Qd8 ... 51 moves. Time: 1/2+0 • BULLET • RATED. 18 hours ago. Opponent: SarcasmGunnel (2728 -1) vs GM Zhigalko_Sergei (3052 +2). Result: White time out • Black is victorious.
- Game 2:** E12 Queen's Indian Defense: Kasparov Variation. 1. d4 e6 2. c4 Nf6 3. Nf3 b6 ... 44 moves. Time: 1+0 • BULLET • CASUAL. 21 hours ago. Opponent: GM Zhigalko_Sergei (3052) vs Serega_Lopuh (2380). Result: Black time out • White is victorious.
- Game 3:** B22 Sicilian Defense: Alapin Variation. 1. e4 c5 2. c3 Nc6 3. d4 cxd4 ... 38 moves. Time: 1+0 • BULLET • CASUAL. 21 hours ago. Opponent: Vovasaiber (1888) vs GM Zhigalko_Sergei (3052). Result: White resigned • Black is victorious.
- Game 4:** B01 Scandinavian Defense: Marshall Variation. 1. e4 d5 2. exd5 Nf6 3. d4 Nxd5 ... 25 moves. Time: 1+0 • BULLET • CASUAL. 21 hours ago. Opponent: GM Zhigalko_Sergei (3052) vs Serega_Lopuh (2380). Result: Black time out • White is victorious.

https://lichess.org/@/Zhigalko_Sergei/search

The screenshot shows the Lichess website interface for a user search. The main header includes the Lichess logo and navigation links: PLAY, PUZZLES, LEARN, WATCH, COMMUNITY, TOOLS. The user profile for GM Zhigalko_Sergei is displayed, showing a rating of 2537 (Rank 3) for ULTRABULLET, 3107 (Rank 4) for BULLET, 2839 (Rank 76) for BLITZ, 3007 (Rank 1) for RAPID, 0 for CLASSICAL, and 2457 (Rank 23) for CRAZYHOUSE. The search filters are set to: Date (mm/dd/yyyy), Rating (From/To), Number of turns (From/To), Duration (From/To), Clock initial time (From/To), Clock increment (From/To), Source, Variant, and Mode. The search results table shows 84,898 games for the selected filters. The bottom of the page shows the number of friends online and the current time (3:54).

На картинке (выше) lichess само предоставляет полный вариант (структуру) возможных запросов.

Вопрос: - можно вот это полностью интегрировать / скопировать в свою базу ?

Как это всё предполагается использовать :

Как платформу для оценки эффективности алгоритмов прогнозирования.

Пример (по смыслу происходящего) сайтов:

<https://www.codingame.com/ide/puzzle/onboarding>

<https://russianaicup.ru/>

Пояснения :

- есть какая-то задача (например - игра простейшая), в нашем случае - анализ шахматных партий (ожидаемого результата партии)

- дальше есть какие-то условия, правила, пояснения к игре

- и самое главное - возможность прямо тут же предложить свой код к этой игре (решению этой задачи)
- ещё момент (API для нас не является самоцелью, это как один из методов - например передачи этих алгоритмов), можно вот как здесь сделать - прям код заливается на сайт.
- принципиальный момент "Choose a programming language. You can choose a language you already know" - возможность писать алгоритмы на любых языках.

Отсюда уже проще объяснить требования к базе, ожидаемую нагрузку и скорость работы.

- примерно будем ориентироваться на 100 одновременно работающих алгоритмов оценки

- оценка должна производиться максимально быстро: идут турниры/партии в режиме он-лайн. Часто время от "жеребьёвки"/появления расписания на следующие партии - это буквально несколько секунд. А иногда и бывает вообще - жеребьёвка и через секунду партия.

- вот за это время (читай секунду) - алгоритм должен предоставить оценку вероятности

- при этом рассчитываем на 100 алгоритмов

- ну и так-же с запасом рассчитываем на 100 партий

- получаем : ежесекундно база (программный комплекс) должен будет обрабатывать и выдавать $100 \times 100 = 10.000$ оценок

P.p.s. "опциональные" задачи пока не расписываются, сюда будут входить:

турниры

- <https://lichess.org/tournament>

- <https://lichess.org/swiss>

выборка игр по игрокам, по игрокам со "статусом" (например GM - гроссмейстер, IM - международный мастер и т.д)

по стримерам <https://lichess.org/streamer>,

трансляциям <https://lichess.org/broadcast>,

и т.д.

(в базовом прототипе - не требуется).

(Один из возможных вариантов архитектуры.):

1. База (с обновлением в режиме реального времени).
2. Программа, которая обращается к базе.
3. Программа, которая обращается к программе, которая обращается к базе:
 - несёт в себе алгоритм
 - получает на вход массив данных
 - обрабатывает алгоритм
 - также является шаблоном для пользователей (разработчиков) алгоритмов

 - результатом работы которой будет оценка вероятности исхода партии (партий)
4. Система сборки, которая принимает программу №3, тестирует, запускает.
5. Система контроля/слежения (менеджер одновременного запуска, проверка рабочего состояния алгоритмов (программ №3))

6. Программа, которая работает с полученными результатами (собирает, оценивает, сравнивает друг с другом).