**Битва интеллектов -2022**

**Интеллектуальная обучающая система**

**по решению задач по математике**

**для школьников**

название проекта

**Авторы проекта:**

**Руководитель проекта:**

**Консультант проекта:**

**Москва – 2022**

**Содержание**

Аннотация 3

1: Цель и задачи 4

2: Актуальность 4

3: Имеющиеся решения данной проблемы 4

4: Преимущества предлагаемого решения 5

5: Обзор технологий построения 5

6: Выбор технологии программирования 6

7: Интерфейс пользователя 8

8: Проект системы 9

9: Типовые действия пользователя 12

10: Выводы и результат 13

11: Список использованных источников 14

Приложение 15

**Аннотация**

Целью нашего проекта является создание обучающей системы с использованием интеллектуальной рекомендательной системы.

В обучающей системе на основе результатов предварительного тестирования ученика выявляются неверно понятые темы, и производится выборка задач для изучения задач по схожей тематике.

В этой системе есть модуль начального тестирования, модуль подбора задач и модуль контрольного тестирования.

Созданная система может быть использована как в условиях школьного обучения, так и самостоятельного изучения

разделов математики по школьному курсу за 7 класс.

1. **Цель и задачи**

Наш проект предназначен для учеников 7-х классов средней школы.

Цель данный работы заключается в создании программной системы, которая будет тестировать знания учеников по алгебре и геометрии, и при неспособности ученика решить какое-либо задание, будет выдавать материал для повторения недостаточно изученных знаний.

Разработанная система может помочь учителям тестировать знания учеников и вести учет ошибок совершенных учениками во время тестирования.

**2. Актуальность**

Проблема своевременного определения качества усвоенного материала актуальна в наше время, ведь учителя не всегда могут уделить время всем ученикам и следить за знанием всего предмета и понимании его учениками. Разработанный сервис поможет решить эту проблему с помощью тестирования всех учеников и выдачи списка задач персонально каждому ученику.

**3. Имеющиеся решения данной проблемы**

Мы потратили достаточно большое время на поиск подобной системы, но видимо, их крайне мало. Однако в результате опроса наших знакомых удалось выяснить, что система оценки и рекомендации задач актуальна и востребована у многих учеников в нынешнее время.

**4. Преимущества предлагаемого решения**

Наше решение в виде сервиса будет не только тестировать знания учеников, но также выдавать задачи для улучшения знаний в темах, которые ученик знает неуверенно.

В нашем приложении будет предлагаться ученику фиксированное количество задач по выбранным темам. Учитель имеет право регулировать количество задач предоставляемых ученику и определять их сложность. Учитель может настроить, например, выдачу 25 задач 3го уровня сложности. Такое задание по моей субъективной оценке является самым высоким уровнем сложности. Или настроить выдачу определённого количества задач каждого уровня. Поэтому данный подход к тестированию является методически логичным, ведь всегда можно настроить все под знания класса и предпочтения или знания каждого ученика в отдельности.

Наше приложение находится на сайте куда может заходить как ученик, так и учитель. Учитель создаёт тест и выдаёт его ученику, а тот в свою очередь решает данный тест.

**5. Обзор технологий построения**

Данная программа является клиент- серверным приложением.

Для реализации нашего проекта использовался язык Python и фреймворк Django. Python - один из современных высокоуровневых языков программирования, который используют многие профессиональные программисты. Основной причиной выбора этого языка для создания обучающей программы было знание того, что язык Python крайне быстро развивается и в ближайшем будущем он будет актуален.

Для создания нашего проекта использовались библиотеки Django, Survey.js и Experta для Python.

Разработанное решение проблемы предусматривает педагогическое мнение. Разработанная в проекте система применима как для проверки математических задач, так и может быть настроена для работы с задачами по физике, химии, информатике.

Веб-приложение размещается на сайте, на который может заходить как ученик, так и учитель. Учитель создаёт задания и присылает ученику ссылку, а ученик, переходя по этой ссылке, решает задачи. После решения задач учеником учитель получает доступ к результатам тестов. Результаты тестов хранятся на сайте и всегда доступны учителю и ученику.

**6. Выбор рекомендательной системы**

Коллаборативная фильтрация (или совместная фильтрация) — это один из методов построения прогнозов (рекомендаций) в рекомендательных системах, использующий известные предпочтения (оценки) группы пользователей для прогнозирования неизвестных предпочтений другого пользователя. Его основное допущение состоит в следующем: те, кто одинаково оценивал какие-либо предметы в прошлом, склонны давать похожие оценки другим предметам и в будущем. Например, с помощью коллаборативной фильтрации математическое приложение способно прогнозировать уровень задач которые ученик не в состоянии решить, имея неполный список его знаний (часов уделённых определённой теме по алгебре или геометрии, а также какие темы были усвоены недостаточно хорошо). Прогнозы составляются индивидуально для каждого пользователя, на основе информации собранной от предыдущих участников. Тем самым коллаборативная фильтрация дает усреднённую оценку для каждого объекта интереса, к примеру, базирующуюся на количестве присланных правильных ответов.

Прогно́з (от [греч.](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B5%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) [πρόγνωση](https://ru.wiktionary.org/wiki/%CF%80%CF%81%CF%8C%CE%B3%CE%BD%CF%89%CF%83%CE%B7) «предвидение, [предсказание](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D1%81%D0%BA%D0%B0%D0%B7%D0%B0%D0%BD%D0%B8%D0%B5)») — это научно обоснованное суждение о возможных состояниях объекта в будущем и (или) об альтернативных путях и сроках их осуществления. В узком смысле, это вероятностное суждение о будущем состоянии объекта исследования.

Существует два основных метода, используемых при создании рекомендательных систем — коллаборативная фильтрация и контентно-основанные рекомендации. Коллаборативная фильтрация, в свою очередь, также разделяется на два основных подхода (типа).

**Основанный на соседстве**

Этот подход является исторически первым в коллаборативной фильтрации и используется во многих рекомендательных системах. В данном подходе для активного пользователя подбирается подгруппа пользователей схожих с ним. Комбинация весов и оценок подгруппы используется для прогноза оценок активного пользователя. У этого подхода можно выделить следующие основные шаги:

* Присвоить вес каждому пользователю с учётом схожести его оценок и активного пользователя.
* Выбрать несколько пользователей, которые имеют максимальный вес, то есть максимально похожи на активного пользователя. Данная группа пользователей и называется соседями.
* Высчитать предсказание оценок активного пользователя для неоценённых им предметов с учётом весов и оценок соседей.

**Основанный на модели**

Данный подход предоставляет рекомендации, измеряя параметры статистических моделей для оценок пользователей, построенных с помощью таких методов как, [метод байесовских сетей](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D0%B5%D1%81%D0%BE%D0%B2%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), [кластеризации](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7), [латентной семантической модели](https://ru.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B5%D0%BD%D1%82%D0%BD%D0%BE-%D1%81%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7), такие как [сингулярное разложение](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D0%B3%D1%83%D0%BB%D1%8F%D1%80%D0%BD%D0%BE%D0%B5_%D1%80%D0%B0%D0%B7%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), [вероятностный латентный семантический анализ](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D1%80%D0%BE%D1%8F%D1%82%D0%BD%D0%BE%D1%81%D1%82%D0%BD%D1%8B%D0%B9_%D0%BB%D0%B0%D1%82%D0%B5%D0%BD%D1%82%D0%BD%D0%BE-%D1%81%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7) и [марковский процесс принятия решений на основе моделей](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D0%BA%D0%BE%D0%B2%D1%81%D0%BA%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_%D0%BF%D1%80%D0%B8%D0%BD%D1%8F%D1%82%D0%B8%D1%8F_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B9" \o "Марковский процесс принятия решений). Модели разрабатываются с использованием интеллектуального анализа данных, алгоритмов машинного обучения, чтобы найти закономерности на основе обучающих данных. Число параметров в модели может быть уменьшено в зависимости от типа с помощью [метода главных компонент](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%B3%D0%BB%D0%B0%D0%B2%D0%BD%D1%8B%D1%85_%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82). Необходим компромисс между точностью и размером модели, так как можно потерять полезную информацию в связи с сокращением моделей.

**Гибридный**

Данный подход объединяет в себе подход основанный на соседстве и основанный на модели. Гибридный подход является самым распространённым при разработке рекомендательных систем для коммерческих сайтов, так как помогает преодолеть ограничения изначального оригинального подхода (основанного на соседстве) и улучшить качество предсказаний. Этот подход также позволяет преодолеть проблему разреженности данных и потери информации. Однако данный подход сложнее и дороже в реализации и применении.

**7. Интерфейс пользователя**

На сайте учитель создаёт тест и получает ссылку на прохождение данного теста. Ученик проходит выданный тест и получает домашнее задание по задачам, в которых были допущены ошибки. Учитель может следить за статистикой решенных задач учеников, и выбирать какие задачи будут предоставляться ученикам.

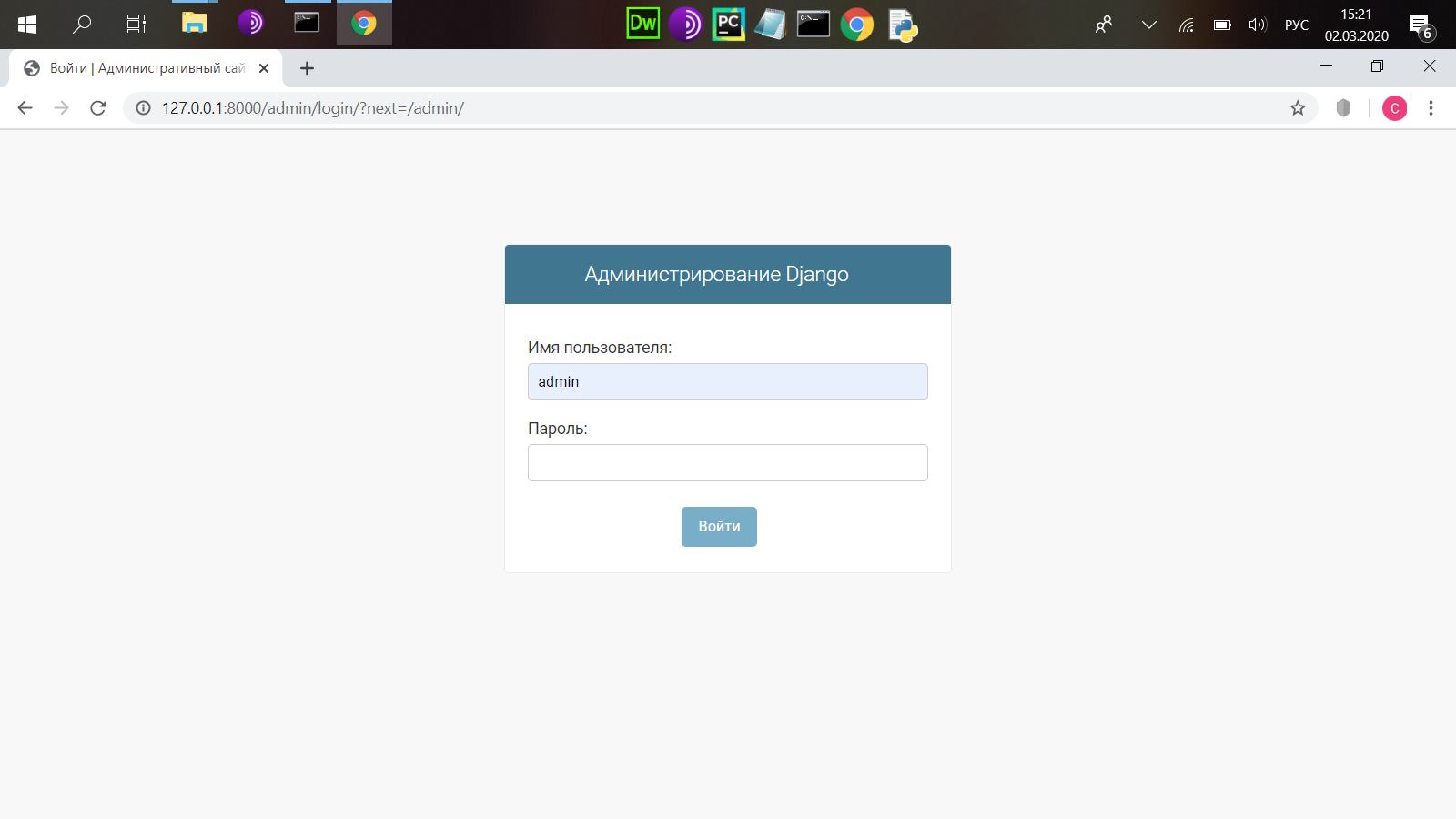


Рисунок 1 – Интерфейс учителя (администратора системы)

**8. Проект системы**

Здесь предоставлены фрагменты блок схемы алгоритма программы тестирования. Детальная блок-схема приведена в приложении 1.

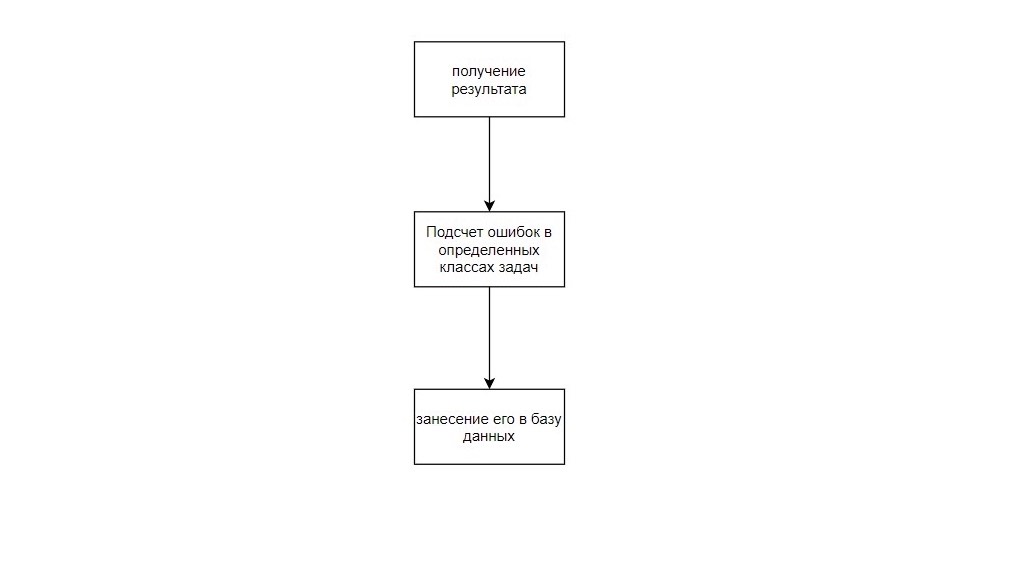


Рисунок 2 – Блок-схема общего алгоритма работы учителя

После повторного тестирования меняется полезность выдаваемых задач в качестве домашнего задания (если задача помогла ее полезность увеличивается, если нет - уменьшается).

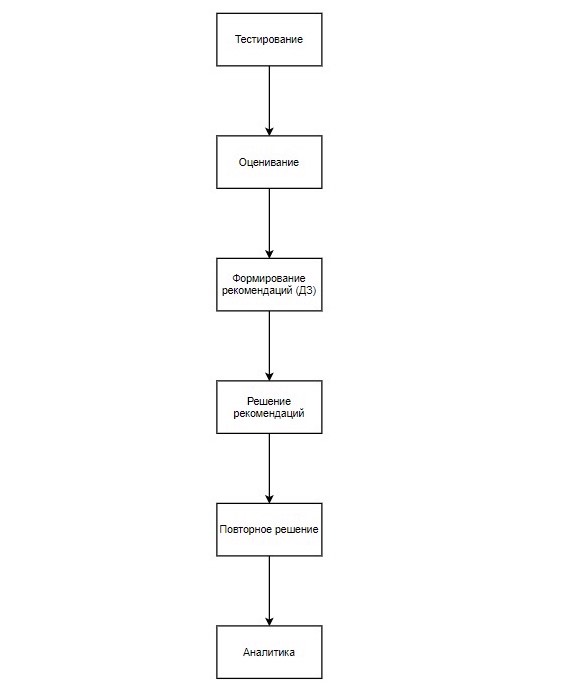


Рисунок 3 – Блок-схема алгоритма модуля формирования оценок и рекомендаций

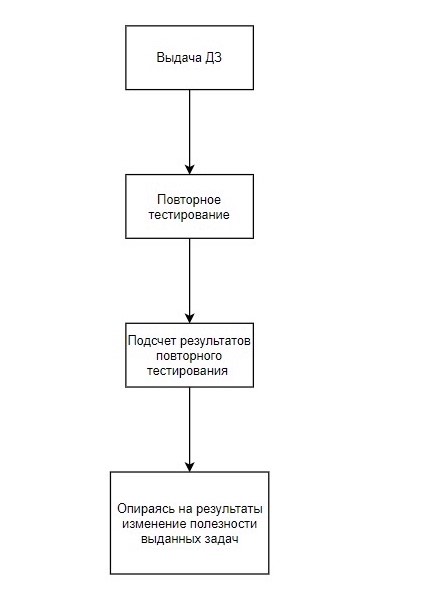


Рисунок 4 – Блок-схема алгоритма модуля тестирования

Формирование блока задач для ДЗ происходит по определённому алгоритму:

при 1-2 ошибках выдаётся 5 задач,   
при 3-4 ошибках выдаётся 10 задач,   
при 5-7 ошибках выдаётся 15 задач,  
если ошибок более 7, то выдаётся 20 задач.

Сам тест состоит из 15 вопросов, которые берутся из перечня всех вопросов введенных учителем. Помимо количества вопросов учитель может вводить сложность выдаваемых задач.

В этом ранжировании и есть отличительная особенность программы, ведь в похожих рекомендательных системах нет возможности выбора уровня сложности предоставляемых задач и их количества.

В текущей конфигурации задачи выдаются по следующему принципу 40% задач по алгебре, 30% задач по геометрии и 30% задач по логике различных уровней сложности.

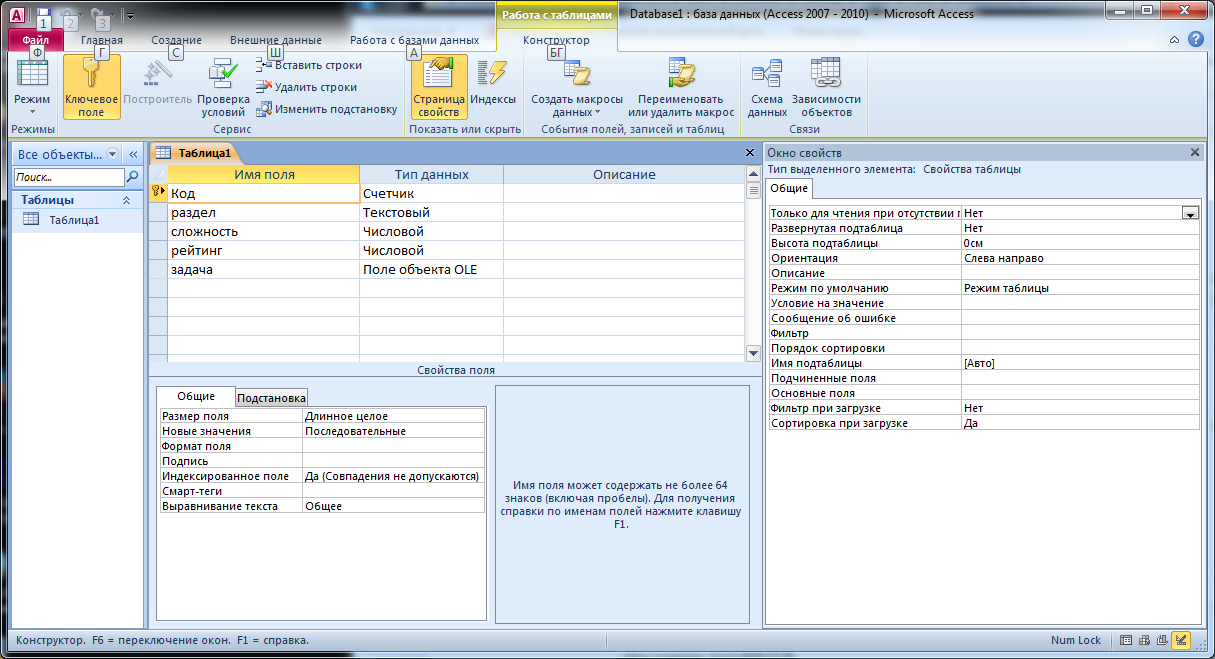


Рисунок 5 – Структура базы данных задач

**9. Типовые действия пользователя**

Пользователю (ученику) выдаётся набор задач, которые ему нужно решить. Для их проверки пользователю требуется ввести ответы в специальную форму. После этого программа или учитель проверяет ответы пользователя, и при наличии ошибок выдаёт домашнее задание по данной теме. Далее система предлагает ученику их решить. После этого проводится контрольное тестирование, и система понимает какие пропуски в знаниях у него есть, и какого типа задачи ему нужно решать.

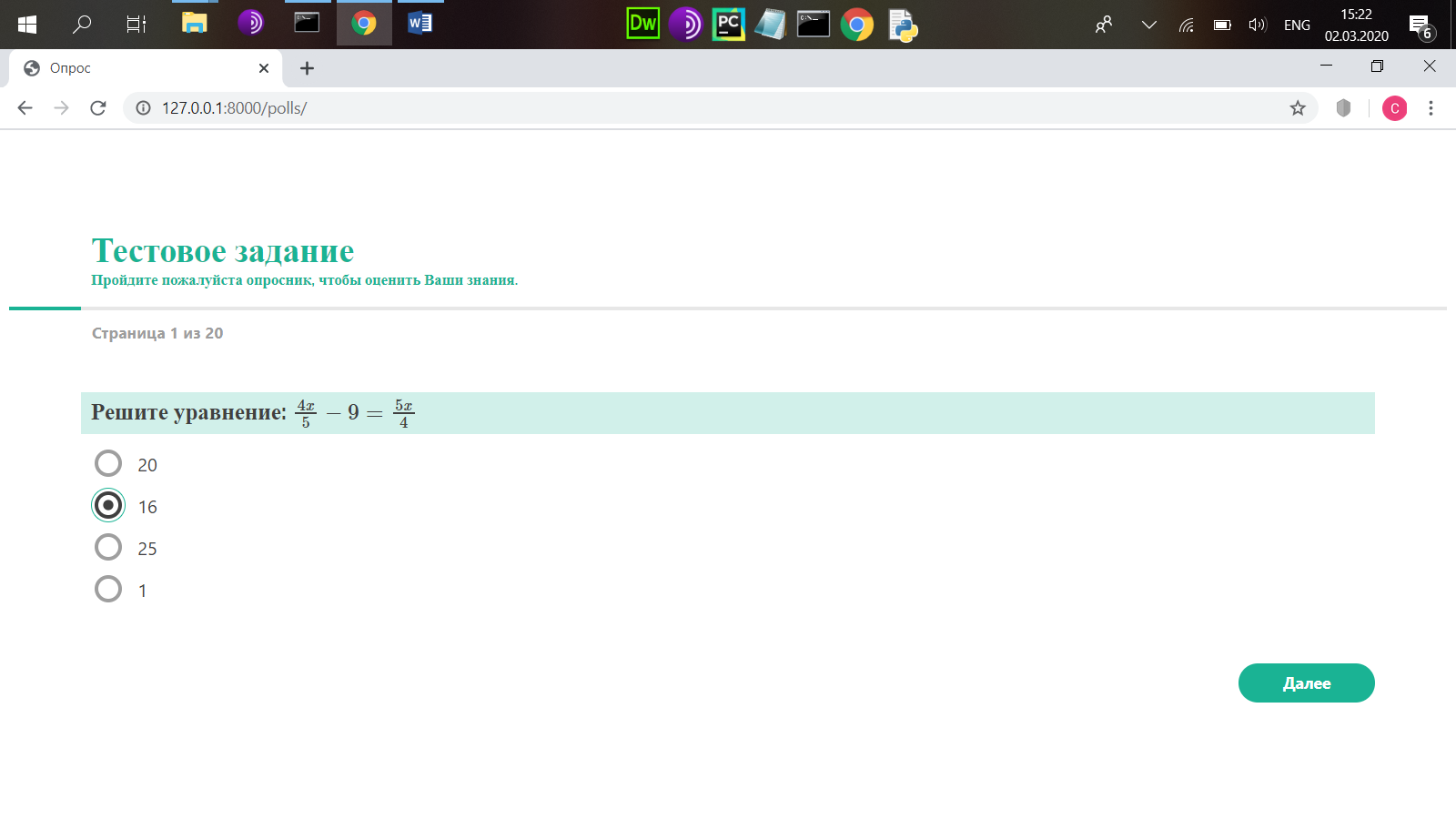


Рисунок 6 – Пример тестового задания

**10. Выводы и результаты**

Созданная интеллектуальная обучающая система с рекомендательным сервисом реализована в виде интернет-сервиса. Система тестирует учеников и реагирует по-разному на различные итоги тестирования. Система ведет учет ошибок и сохраняет их в отдельной базе данных. После каждого тестирования в системе изменяется полезность задач, которые выдаются в качестве домашнего задания.

**11. Список использованных источников**

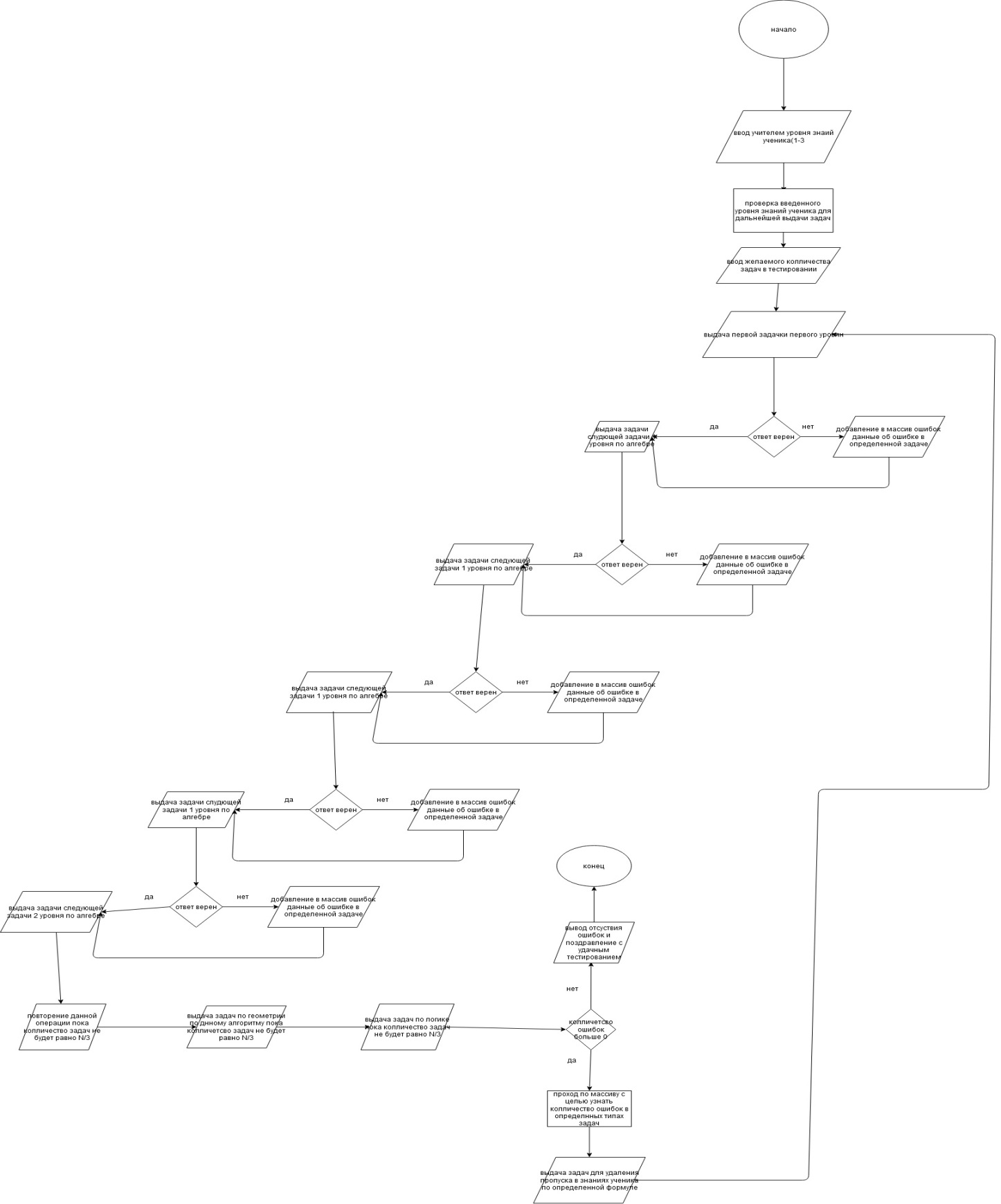
**1.** Язык Python [Электронный ресурс] URL: http://kpolyakov.spb.ru/school/ probook/python.htm

**2.** Объектно-ориентированный анализ и проектирование с примерами приложений. Буч Г., Максимчук Р., Энгл М., Янг Б. Коналлен Д., Хьюстон К. - 3-е изд.: М.:ООО «ИД Вильямс», 2017 – 720 с.

3. Ройзнер, Михаил. «Как работают рекомендательные системы. Лекция в Яндексе»: [Электронный документ] URL: https://habr.com/company/yandex /blog/241455/

**4.** Библиотека Джанго **-** https://ru.wikipedia.org/wiki/Django

**Приложение 1**



*Блок-схема алгоритма тестирования*