

Шубинкин Владимир Николаевич

магистр, учитель

МАОУ «Лицей №131»

г. Казань, Республика Татарстан

DOI 10.31483/r-149776

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ОБРАЗОВАТЕЛЬНОЙ ПЛАТФОРМЫ STEPİK ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ В ШКОЛЕ

***Аннотация:** в статье рассматриваются функциональные возможности образовательной платформы Stepik в аспекте её применения при обучении программированию в школе.*

***Ключевые слова:** информатика, программирование, Stepik, курсы.*

Преподавая программирование в школе, я самостоятельно рассказываю теорию и создаю подборки задач. Эта необходимость обусловлена тем, что учащиеся физико-математического лицея способны усвоить больше информации, чем предлагается в стандартном учебнике, даже при базовом количестве часов, отведённых на информатику.

Учащиеся могут пропускать уроки по разным причинам: болеют, участвуют в олимпиадах и конференциях, семейные обстоятельства и прочее. Возникает проблема: где взять теоретический материал и задачи для тренировки? Набор практических заданий по информатике можно выслать сразу. С теорией же возникают сложности. В Интернете множество сайтов, посвящённых программированию, но недостаточно знать название темы, чтобы найти подходящую информацию. А при внимательном рассмотрении оказывается, что ресурсов, предоставляющих материал в нужном качестве и количестве практически нет (неполнота или переизбыток информации, плохая проработанность задач, отсутствие образцовых решений, примитивность или излишняя сложность изложения). Даже лучшие (по версии Stepik Awards 2020 [5]) курсы по программированию

имеют свои недостатки. Например, курсы «Поколение Python: курс для начинающих» [3] и «Поколение Python: курс для продвинутых» [4] слишком объёмны для школьной программы по информатике. Наличие указанной проблемы сподвигло меня создать собственный курс по основам программирования в соответствии с моими уроками. В качестве платформы был выбран сайт stepik.org, функциональным преимуществам которого посвящена эта статья. Остановимся подробнее на них.

1. Хостинг.

Stepik позволяет создавать и размещать курсы бесплатно. Есть возможность продавать созданные курсы. В этом случае платформа берёт комиссию с продаж.

2. Разнообразие типов задач.

На данный момент доступно создание двух видов шагов с теорией: текст и видео, – и более десятка типов задач: тест, сопоставление, задача на программирование и пр. Во всех задачах, доступных в бесплатном курсе, реализована автоматическая проверка решений.

3. Отслеживание активности учащихся.

Преподаватели курса могут просматривать каждое сданное решение. Кроме того, платформа позволяет создавать Классы (в том числе на чужих курсах) для отслеживания успеваемости всей группы.

4. Комментарии и решения.

В каждом шаге есть раздел комментариев, в котором учащиеся могут поделиться своими мыслями или задать вопрос. В практических шагах присутствует раздел решений, в котором учащиеся могут поделиться своим решением и изучить чужие. В хороших курсах в этом разделе закреплены эталонные решения, с которыми полезно ознакомиться учащимся. По умолчанию раздел с решениями доступен только после того, как система примет собственное решение учащегося.

5. Инструменты форматирования.

Для информативного, удобного для чтения отображения формул (рис. 1) можно воспользоваться встроенным редактором на основе TeX [1] (рис. 2).

Базельская задача

Вводится натуральное число n . Вычислить сумму n первых членов ряда

$$1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots$$

Примечание. Точное значение суммы этого ряда равно $\frac{\pi^2}{6}$

Рис. 1. Отображение формул

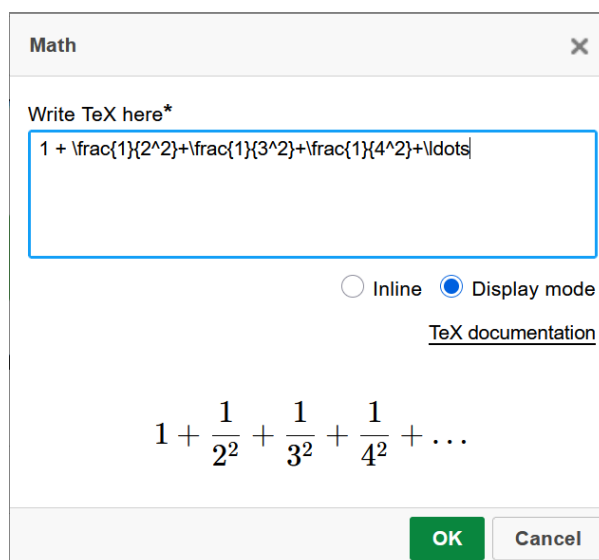


Рис. 2. Интерфейс редактора формул

Для вставок программного кода (рис. 3) имеется свой редактор (рис. 4), позволяющий выбрать используемый язык программирования для правильной подсветки кода.

Язык программирования Python позволяет обходиться без флагов в ситуациях, похожих на описанную выше. Для этого можно использовать `else`, относящийся к циклу (не к условному оператору!).

```
a = int(input())
for x in range(1, 1001):
    if not (x % a == 0 or x % 6 != 0 or x % 4 != 0):
        print('Нет')
        break
else:
    print('Да')
```

В таком случае блок кода внутри `else` выполнится, если цикл завершился **без** `break`.

Рис. 3. Пример текста с вставкой программного кода

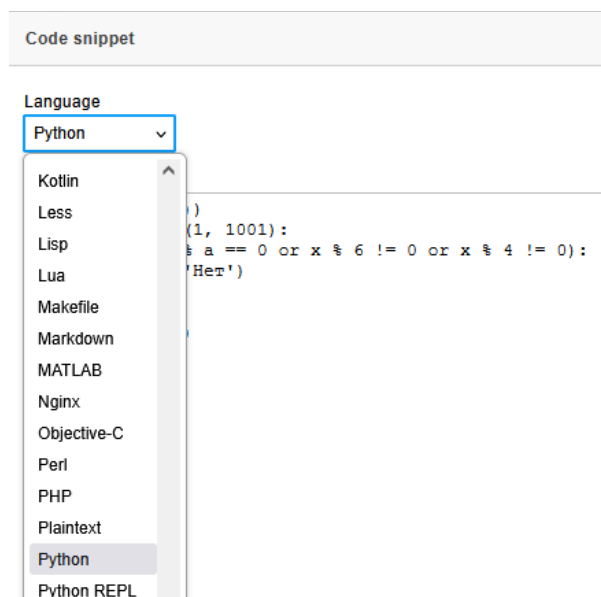


Рис. 4. Интерфейс редактора вставки кода

Также допустима вставка изображений и таблиц, а также базовые возможности форматирования текста: выделение, стили, цвет текста, нумерованные и маркированные списки, выравнивание текста и вставка ссылок. Для большей гибкости есть редактор HTML (рис. 5), позволяющий использовать весь арсенал гипертекстовой разметки. В качестве примера на рис. 5 показано создание раскрывающейся подсказки.

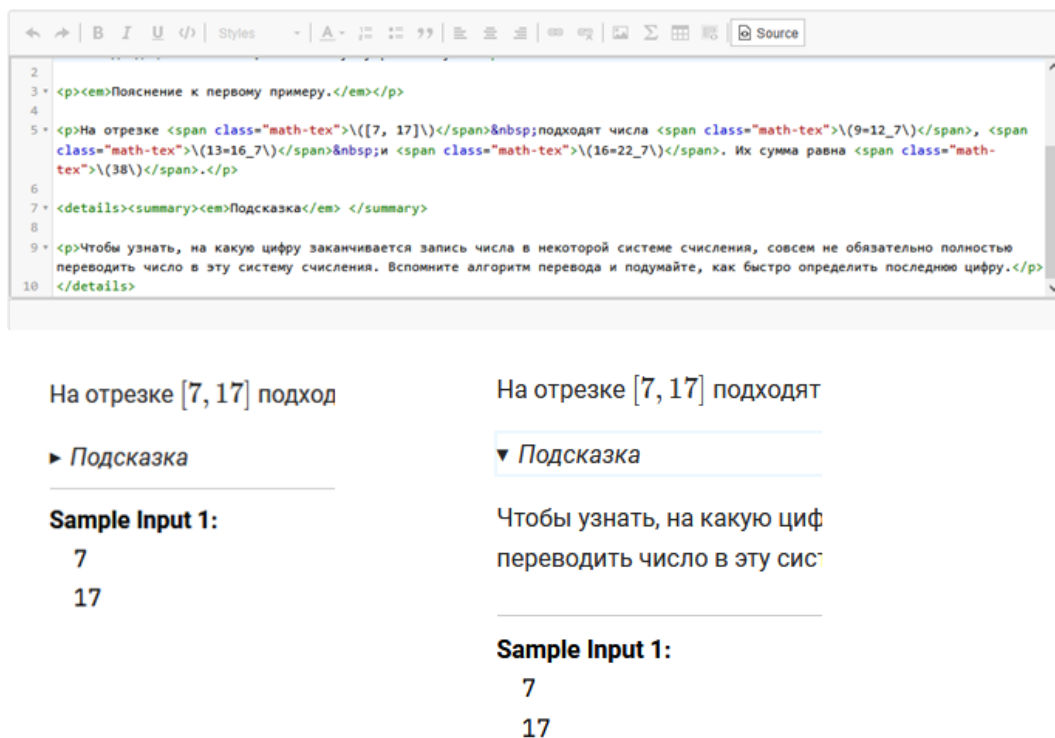


Рис. 5. Редактор HTML

6. Дополнительные возможности редактора заданий.

В заданиях на написание кода можно выбрать язык программирования (из доступных), а также написать код на Python, который будет выполнен до и/или после кода учащегося (рис. 6).

Шаблоны кода и допустимые языки. По умолчанию поле пустое.

```

1 ::python3.12
2
3 ::header
4 a, b = map(lambda s: int(s[2:]), input().split())
5
6 ::footer
7 print(f'a={a}', f'b={b}')
8

```

Рис. 6. Шаблоны кода и языки программирования, доступные учащемуся

В частности, благодаря этому инструменту можно принимать на проверку только реализацию функций без их вызова.

Программы проверяются на наборах тестовых данных и Stepik позволяет генерировать их автоматически (рис. 7).

Введите ваш проверяющий код на Python 3.6.

```

5 def generate():
6     data = ['-3\n']
7     x = -3
8     while x <= 3:
9         x += 0.2
10        data.append(f'{x:.1}\n')
11    return data
12
13 def check(reply, clue):
14     return f'{float(reply):.3}' == f'{float(clue):.3}'
15
16 def solve(dataset):
17     x = float(dataset)
18     if x <= -2:
19         return '5'
20     if x >= 0:

```

Рис. 7. Автоматическая генерация тестовых данных

Функция `generate()` создаёт список входных данных, функция `solve()` на их основе получает верный ответ, а функция `check()` сравнивает ответ, полученный программой учащегося, с ответом, полученным функцией `solve()`. Следует отметить, что проверка ответов настраивается достаточно гибко. Например, на рис. 7 значения сравниваются с точностью до трёх знаков после запятой.

В различных типах заданий присутствует возможность выдать комментарий к правильному и неправильному решению (рис. 8).

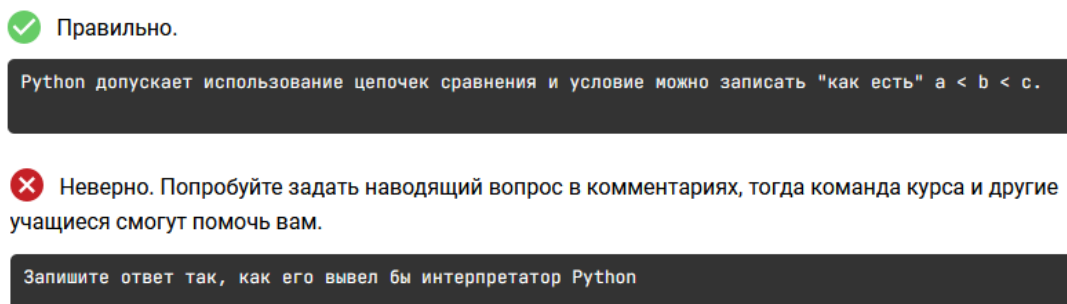


Рис. 8. Примеры комментариев к решениям учащихся

Таким образом, платформа Stepik позволяет бесплатно создавать собственные курсы, содержащие теорию и всевозможные типы практических задач с автоматической проверкой. При этом учитель может отслеживать прогресс класса в целом и просматривать каждое отдельное решение, чтобы дать обратную связь. Благодаря большому набору инструментов форматирования, конспекты и условия задач выглядят удобнее для их прочтения. Гибкие настройки позволяют генерировать тестовые данные автоматически; сверять ответ с правильным по собственным критериям; добавлять код, который выполнится до и/или после кода учащегося. Но эти настройки доступны только на языке программирования Python и часть из них работает только для программ, написанных на Python. Кроме того, отсутствует возможность запрета использования определённых инструментов. Например, использование массивов, когда в этом нет необходимости. В качестве минуса можно отметить зависимость от стабильности Интернет-соединения и работы серверов Stepik.

В 2024–2025 учебном году мною создан курс по основам программирования для 8 класса [2] и апробирован на группе учащихся. Как результат проекта, отмечается повышение мотивации учащихся по программированию на языке Python. Поэтому данный курс будет расширяться темами 9–11 классов. Платформу Stepik можно рекомендовать к использованию в школе на уроках информатики, так как перечисленные функциональные возможности дают необходимые условия для оптимальной работы как учителя, так и его учеников.

Список литературы

1. Документация TeX [Электронный ресурс]. – Режим доступа: <https://en.wikibooks.org/wiki/LaTeX/Mathematics> (дата обращения: 20.06.2025).
2. Шубинкин В.Н. Курс «Информатика. Уроки Владимира Николаевича» / В.Н. Шубинкин // Stepik [Электронный ресурс]. – Режим доступа: <https://stepik.org/course/219675/promo> (дата обращения: 20.06.2025).
3. Гуев Т.А. Курс «Поколение Python: курс для начинающих» / Т.А. Гуев // Stepik [Электронный ресурс]. – Режим доступа: <https://stepik.org/course/58852/promo> (дата обращения: 20.06.2025).
4. Гуев Т.А. Курс «Поколение Python: курс для продвинутых» / Т.А. Гуев // Stepik [Электронный ресурс]. – Режим доступа: <https://stepik.org/course/68343/promo> (дата обращения: 20.06.2025).
5. Сертификат Stepik Awards 2020 в номинации «Лучший бесплатный курс» [Электронный ресурс]. – Режим доступа: <https://ucarecdn.com/5cf1b9b8-5df0-47e1-a977-05642ef9d293/> (дата обращения: 20.06.2025).