АВТОГЕНЕРАЦИЯ ПРОВЕРОЧНЫХ ТЕСТОВ ДЛЯ КОНТРОЛЯ ФОРМИРОВАНИЯ НАВЫКОВ ВЫПОЛНЕНИЯ РАСЧЕТНО-ГРАФИЧЕСКИХ ЗАДАЧ

Щербина Д.Н.
ФГАОУ ВО «Южный федеральный университет», Учебно-научно-исследовательский институт биомедицинских информационных технологий
E-mail: dnsherbina@sfedu.ru

Навыки выполнения расчетно-графических задач приобретались студентами-биологами при освоении курса «Математические методы в биологии». Поскольку большинство биологов не склонны к глубокому пониманию алгоритмов программирования, то был выбран подход, где искомый алгоритм решения расчетно-графической задачи собирается из отдельных шагов, для которых предоставлены готовые образцы кода. Образцы кода на языке Python, снабженные комментариями и пояснениями, предоставлялись студентам в виде электронных блокнотов в формате Jupyter Notebook. Электронные блокноты были выложены в интернет-репозитории, поэтому могли быть просмотрены с помощью сервиса http://nbviewer.jupyter.org (например, через браузер мобильного устройства). В этом случае блокноты с образцами кода выступали в качестве справочника. На практических занятиях и дома студенты скачивали блокноты на рабочий компьютер для формирования навыков решения конкретных задач путем непосредственной работы с кодом в ячейках блокнота. Поскольку образцы кода были снабжены подробными комментариями и ссылками на дополнительные источники сведений, то в данном случае содержащие их блокноты могли выступать в качестве учебно-методического пособия для самостоятельной работы.

После освоения конкретных приемов обработки и визуализации данных в рабочей среде Jupyter Notebook, студенты были готовы к проверке приобретенных навыков. Автоматизированный контроль был реализован в виде теста на образовательном портале из нескольких заданий, за решение которых студенты могли получить баллы. Рекомендованный способ решения заданий заключался в создании нового блокнота, в котором с помощью заголовков 1-2 уровня задавалась структура для удобной навигации. Для каждого задания теста рекомендовалось скопировать и вставить условие задачи, далее создать ячейки с кодом, в которых задать необходимые переменные в соответствии с условием задачи. Для лучшего понимания необходимо было добавить комментарии о последовательности необходимых операций для решения задачи. Далее в учебных материалах нужно было найти образцы кода, с

помощью которых можно выполнить необходимые операции. После преобразования кода так, чтобы он правильно выполнял нужные операции в правильной последовательности, студент получал требуемый ответ. Когда все ответы получены и зафиксированы в рабочем блокноте, студенту остается только внести их в соответствующие поля формы теста на образовательном портале. Рабочий блокнот с найденными решениями и собственными комментариями студент может включить в свое портфолио.

Теоретически правильные ответы можно было получить и другими способами. Например, некоторые студенты пытались решить задачу, опираясь на имеющиеся навыки работы в программе MS Excel. Однако, проверочные задания были составлены таким образом, что получить баллы с наименьшими трудозатратами можно было путем решения рекомендуемым способом. Этому способствовали следующие меры:

1. Автогенерация банка уникальных вопросов в количестве, превышающем размер группы, что исключало мотивацию списать у товарища.
2. Для получения ответа необходимо было выполнить 3-5 логических шагов, подкрепленных расчетами, поэтому угадать правильный ответ среди однотипных дистракторов можно было только случайно.
3. В некоторых случаях параметры задачи были привязаны к программным модулям на Python, например, ответ зависел от генератора случайных чисел (ГСЧ) модуля numpy:

```
Инициализируйте ГСЧ с состоянием 555. Сгенерируйте выборку из 20
значений из нормального распределения с параметрами: scale=10,
loc=100. Постройте гистограмму распределения и определите,
сколько значений попали в диапазон от 86.3 до 90.2?
      2
      4
      0
      1
```

4. Предварительное освоение конкретных приемов обработки и визуализации с помощью готовых образцов кода убеждало студентов в их работоспособности и гибкости. Поиск примеров решения во внешних источниках, например, на форумах в интернете, становился нецелесообразным.

Автогенерация заданий проводилась с помощью скрипта, комбинирующего несколько параметров из заданных списков, или формирующего конкретные подвыборки значений для анализа из исходного большого набора значений. Далее каждая созданная в скрипте задача автоматически решалась, подбирались правдоподобные дистракторы, и полученные таким образом вопросы с вариантами ответов

сохранялись в виде текстового файла в формате GIFT. Набор вопросов вручную импортировался в банк заданий образовательного портала в отдельную категорию, на основе которой формировался тест. Достоинством данного подхода является возможность с помощью небольших модификаций скрипта получить новый набор уникальных заданий, например, для студентов следующего курса. Формат блокнота позволяет компактно сохранять необходимые описания, формулы и рекомендации, которые требуются при формировании нового теста.

Таким образом, представленный способ автогенерации тестовых заданий позволяет автоматизировать контроль формирования навыков выполнения расчетно-графических задач, мотивируя при этом студентов на самостоятельную работу по выработке алгоритмического мышления.