

П. А. Хорошевич

P. Khoroshevich

*УО «Белорусский государственный педагогический университет
имени Максима Танка» (Минск, Беларусь)*

РАЗРАБОТКА ОНЛАЙН-СЕРВИСА ДЛЯ АВТОМАТИЧЕСКОЙ ПРОВЕРКИ SCRATCH-ПРОЕКТОВ

DEVELOPMENT OF ONLINE SERVICE FOR AUTOMATING SCRATCH PROJECTS ASSESSMENT

В статье описываются особенности поиска и исправления синтаксических и семантических ошибок в языке программирования Scratch. Дается описание возможностей разработанного нами онлайн-сервиса для проверки Scratch-проектов.

The article describes the specifics of detecting and fixing syntactic and semantic errors in the Scratch programming language. It describes the capabilities of the online service for checking Scratch projects developed by the author.

Ключевые слова: программирование; Scratch; отладка программы; оценивание проекта.

Keywords: programming; Scratch; program debugging; project grading.

Содержательная линия «Основы алгоритмизации и программирования» имеет фундаментальное значение для школьного учебного предмета «Информатика». Поэтому актуальной становится задача поиска эффективных инструментов и методов обучения основам алгоритмизации, а также способствующих развитию алгоритмического мышления учащихся [1]. Одним из таких инструментов является визуализированный язык программирования Scratch.

Программа на языке Scratch состоит из последовательности визуальных блоков, соединённых между собой. Назначение блока, тип параметров и переменных обозначается цветом и формой. Таким образом, соединить между собой можно лишь подходящие по смыслу блоки, что почти исключает возникновение синтаксических ошибок в коде программы. Примерами синтаксических ошибок может послужить объявление собственного блока с несколькими параметрами, имеющими одинаковые имена или использование параметров вне определения собственной процедуры. В Scratch-проекте не исключено появление ошибок, свойственных другим языкам программирования: отсутствие инициализации переменных, бесконечные циклы и бесконечная рекурсия.

Помимо ошибок, код может содержать особенности, снижающие его читаемость: отсутствие комментариев, длинные скрипты, дублирование кода, неинформативные имена спрайтов и переменных, наложение одного скрипта на другой и потерянные блоки. Пользователи платформы Scratch могут реализовать собственную идею на основе существующего проекта другого пользователя. В таком случае ясность и читаемость кода приобретает ещё большее значение [2].

С целью автоматизации поиска синтаксических и семантических ошибок в Scratch-проектах, а также их оценивания нами был разработан онлайн-сервис «КотФикс» [3].

На главной странице сервиса учащийся загружает файл Scratch-проекта или указывает ссылку на страницу проекта в онлайн-среде разработки [4]. По окончании загрузки на странице отображается отчёт с итоговой отметкой и списком **замечаний**: предупреждений и ошибок.

Под **предупреждениями** будем понимать замечания, которые непосредственно не влияют на корректность работы алгоритма, но сказываются на читаемости кода. В данный момент сервис может указать пользователю на следующие проблемы в проекте: отсутствие комментариев, наличие неиспользуемых переменных, наложение скриптов друг на друга, наличие «потерянных блоков», обнаружение спрайтов со стандартными именами, поиск слишком длинных скриптов, обнаружение спрайтов без скриптов.

Вторая категория замечаний – **ошибки**. В замечаниях этого типа описываются аспекты функционирования скриптов, которые могут привести к семантическим ошибкам в работе алгоритма: отправка спрайтом сообщения, которое не получает ни один спрайт, или получение сообщения, которое не отправляет ни один спрайт; использование переменной без указания начального значения и сравнение двух литералов (буквальных значений).

Каждое замечание имеет собственное название и описание. Если замечание относится к конкретному спрайту, указывается его имя. Там, где это возможно, в замечании присутствует фрагмент скрипта, который привёл к появлению предупреждения или ошибки.

Обнаружение и исправление ошибок на раннем этапе работы над проектом помогает избежать формирования у учащихся вредных паттернов программирования, применение которых приведёт к ухудшению качества кода [5].

Scratch-проекты оцениваются сервисом автоматически по следующим критериям: **поток данных** – использование циклов с определённым количеством повторений и циклов с предусловием; **использование переменных** – переменные инициализированы начальными значениями, используются как простые, так и составные типы данных (списки); **логика** – применяется сокращённая или полная форма условного оператора, есть составные условия (операторы И, ИЛИ, НЕ); **параллельность** – одновременный запуск нескольких скриптов по щелчку, нажатию клавиши на клавиатуре или при отправке сообщения; **абстракция** – используется многократный вызов собственных блоков, создаются клоны спрайтов; **синхронизация** – применяется запуск скриптов после получения сообщения или смены фона; **интерактивность** – управление спрайтами с помощью мыши, клавиатуры, веб-камеры или микрофона.

В каждой критерии учащийся получает от 0 до 3 баллов. Результат оценивания содержит как суммарное количество баллов, так и отметку в каждой категории. Учащийся видит условие получения того или иного балла за категорию. Описание критериев оценивания соответствует регламенту международной Scratch-олимпиады по креативному программированию [6].

В заключение продемонстрируем необходимость использования сервиса на примере проверки проектов, размещённых в официальной репозитории Scratch [4].

После проверки 200 случайных проектов из категории «Игры» получилось распределение предупреждений, приведённое на рисунке 1. Для нормализации значений общее количество предупреждений каждого типа делилось на количество спрайтов в проекте. Таким образом мы получили количество предупреждений на 1 спрайт.

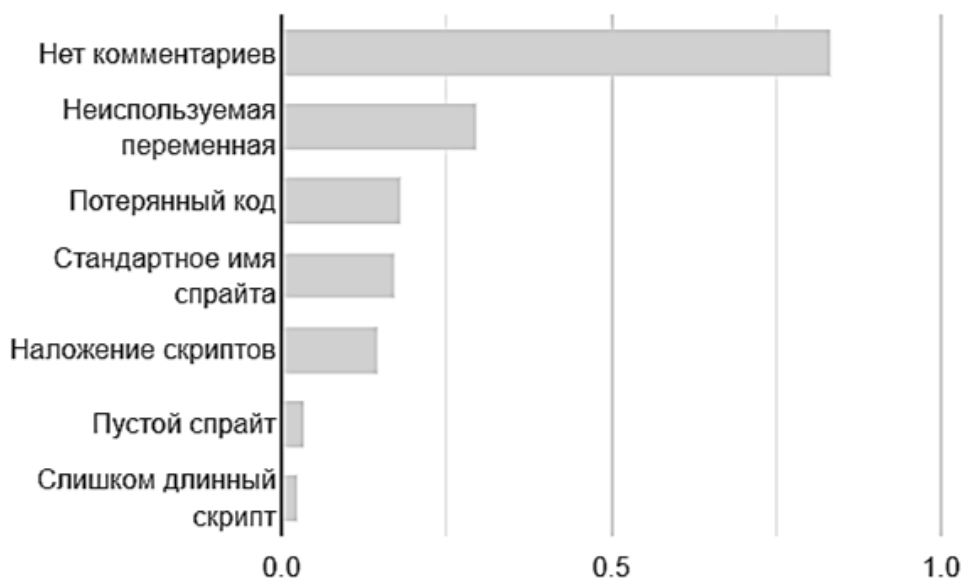


Рисунок 1 – Количество предупреждений на 1 спрайт по результатам проверки Scratch-проектов

Стоит отметить, что замечания присутствовали во всех проектах, 128 проектов из 200 содержали только предупреждения и не содержали ошибок. Исходя из результатов проверки можно сделать вывод о необходимости применения инструментов автоматического оценивания кода на визуализированном языке программирования Scratch.

Список использованных источников

1. Вабищевич, С. В. Базовые акценты изучения учебной дисциплины «Программирование в визуализированных средах» будущими учителями физики и информатики / С. В. Вабищевич, А. А. Францкевич // Вес. БДПУ. Сер. 3, Фізика. Матэматыка. Інфарматыка. Біялогія. Геаграфія. – 2022. – № 4. – С. 37–41.
2. Hermans F. Do Code Smells Hamper Novice Programming? A Controlled Experiment on Scratch Programs / F. Hermans, E. Aivaloglou // 24th International Conference on Program Comprehension. – Austin, 2016. – P. 1-10.
3. Онлайн-сервис КотФикс [Электронный ресурс]. – Режим доступа: <https://kotfix.netlify.app>. – Дата доступа: 26.09.2023.
4. Среда разработки Scratch [Электронный ресурс]. – Режим доступа: <https://scratch.mit.edu>. – Дата доступа: 26.09.2023.
5. Meerbaum-Salant O. Habits of Programming in Scratch / O. Meerbaum-Salant, M. Armoni, M. Ben-Ari // ITiCSE '11: Annual Conference on Innovation and Technology in Computer Science Education. – Darmstadt, 2011. – P. 168-172.
6. Белорусский национальный этап Scratch-олимпиады [Электронный ресурс]. – Режим доступа: <https://creativeprogramming.org/by>. – Дата доступа: 26.09.2023.