

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО РАЗРАБОТКЕ АВТОМАТИЗИРОВАННОГО ПРАКТИКУМА В СРЕДЕ «КУМИР» ДЛЯ ИСПОЛНИТЕЛЯ РОБОТ

Перевозчикова М. С., к.п.н., доцент,

Нечаева В. В., магистрант,

Вятский государственный университет, г. Киров, Россия

Аннотация. В статье представлены методические рекомендации по созданию автоматизированного практикума в учебной среде «КуМир» для исполнителя «Робот». Подробно описан двухэтапный процесс разработки, включающий подготовку файловых ресурсов (защищённых шаблонов программ, начальных обстановок и алгоритмов автоматического тестирования) и последующую верстку структурированного XML-файла курса. Особое внимание уделено техническим требованиям к уникальности идентификаторов, корректности относительных путей, кодировке UTF-8 и процедуре проверки работоспособности практикума. Предложенный алгоритм позволяет педагогам самостоятельно конструировать интерактивные учебные модули с автоматизированной оценкой решений, что способствует эффективной организации самостоятельной работы школьников.

Ключевые слова: автоматизированный практикум, среда «КуМир», исполнитель «Робот», XML-разметка, образовательные ресурсы, обучение программированию.

Среда «КуМир» с исполнителем «Робот» остается ключевым инструментом формирования алгоритмического мышления, однако педагоги часто испытывают дефицит готовых решений для объективного контроля. Использование автоматизированного практикума позволяет снизить рутинную нагрузку на учителя, обеспечить мгновенную обратную связь и адаптацию задач под уровень обучаемого. Наличие методических рекомендаций

критически важно для разработки и интеграции этого инструмента в учебный процесс.

Автоматизированный практикум представляет собой электронный образовательный ресурс, предназначенный для организации самостоятельной работы обучающихся. В среде «КуМир» такой практикум реализуется в виде структурированного XML-файла (с расширением .kurs.xml), который описывает иерархию учебных модулей (разделов) и входящих в них задач, а также связывает каждый модуль с необходимыми файловыми ресурсами: начальными обстановками исполнителя и шаблонами программ. Процесс разработки практикума условно можно разделить на два этапа.

1. Предварительный этап

Прежде чем приступить к написанию XML-кода, необходимо разработать файловое наполнение практикума. От качества этого этапа напрямую зависит работоспособность будущего курса. Файловое наполнение практикума состоит из файлов-задач и файлов-обстановок. Такие файлы создаются в среде «Кумир для учителей». Для каждой задачи создается отдельный файл с расширением .kum, который будет открываться в среде «КуМир» как стартовая программа. Создаваемый файл условно можно поделить на следующие части: шаблон программы и тестирование.

Шаблон программы (рис. 1) имеет следующую структуру:

- строка 1 содержит имя исполнителя и необходима для корректной работы всего программного кода;
- в строке 2 продублирована формулировка задачи. Обратите внимание, что она записана в коде как комментарий (ALT+!);
- в строке 3, кроме слова «алг», указывающего на начало кода, также указывается имя алгоритма – в дальнейшем именно это имя будет вызываться внутри тестирования;
- команды «нач» и «кон».

Написанные строки необходимо защитить от изменений. Для этого необходимо привести курсор слева от номера защищаемой строки и нажать на

появившийся символ замочка. Аналогичным образом выполняется защита всех остальных строк. Стоит отметить, что строки между командами «нач» и «кон» не нужно защищать, поскольку именно в этом промежутке обучающийся будет писать решение задачи.

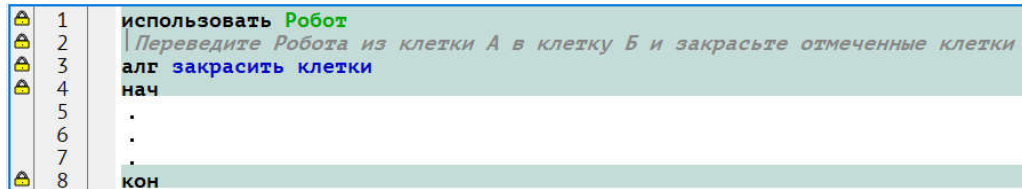


Рисунок 1 – Защищенный от изменений код-шаблон

В процессе создания фрагмента файла (рис. 2), содержащего код для тестирования, необходимо учесть следующие замечания:

- код тестирования пишется в том же файле, что и шаблон;
- имя «алг цел @тестирование» является обязательным, иначе среда программирования не воспримет написанный код как тестирование;
- в первой строке алгоритма (после «нач») обязательно вызывать имя ранее разработанного шаблона, поскольку предполагается, что обучающийся записал свое решение задачи в пустое пространство шаблона;
- далее считывается конечное положение робота после выполнения алгоритма и проверяется, дошел ли робот до клетки, отмеченной буквой Б. Если робот закончил выполнение программы в нужной клетке, то переменной «знач» присваивается значение 10, иначе присваивается значение 2 и на экран выводится сообщение об ошибке;
- также необходимо проверить, закрашены ли все отмеченные клетки и не закрашены ли лишние. В случае ошибочного закрашивания на экране появляется сообщение об ошибке и за задачу выставляется оценка 2;
- если после всех проверок значение переменной «знач» больше 8, то выводится сообщение о правильности выполнения задания.

```

10 алг цел @тестирование
11 нач
12 . закрасить клетки
13 . цел роб строка, роб столбец
14 . @@робот(роб строка, роб столбец)
15 . если @нижняя буква(роб строка, роб столбец)='Б' то
16 . . . знач:=10
17 . . иначе
18 . . . вывод "Ошибка! Робот не пришел в клетку Б!", нс
19 . . . знач:=2
20 . . . выход
21 . все
22 .
23 . цел столбцов, строк
24 . @@размер поля(строк, столбцов)
25 . цел строки, столбцы
26 . нц для строки от 1 до строк
27 . . нц для столбцы от 1 до столбцов
28 . . . если (@метка(строки, столбцы) и не @@закрашена(строки, столбцы))
29 . . . . то
30 . . . . . вывод "Ошибка! Отмеченная клетка не закрашена!"
31 . . . . . знач:=2
32 . . . . . @поставить оценку(2)
33 . . . . . стоп
34 . . . . все
35 . . . если (не @метка(строки, столбцы) и @@закрашена(строки, столбцы))
36 . . . . то
37 . . . . . вывод "Ошибка! Закрашена лишняя клетка!"
38 . . . . . знач:=2
39 . . . . . @поставить оценку(2)
40 . . . . . стоп
41 . . . все
42 . . .
43 . . кц
44 . кц
45 . если знач>8 то
46 . . . вывод "Задание выполнено правильно!"
47 . все
48 кон

```

Рисунок 2 – Алгоритм тестирования задачи

Готовый алгоритм тестирования необходимо скрыть от обучающихся перетягиванием серой пунктирной линии на строку предшествующую коду тестирования. Файлы программ удобнее разместить в отдельной директории, так как путь к ним будет в дальнейшем использоваться. Следует давать задачам осмысленные имена, например, имя файла может отражать номер и/или суть задачи.

Каждая задача должна быть обеспечена несколькими файлами начальной обстановки. Эти файлы хранят конфигурацию поля исполнителя «Робот»: размеры, расположение стен, наличие закрашенных клеток, позиции маркеров А и Б. Обстановка создается с помощью встроенного редактора обстановок среды «КуМир» и сохраняется в файле формата *.fil. Для работы с обстановками нужно убедиться, что внутри программной среды открыто соответствующее окно.

В пустой обстановке необходимо построить стены, согласно требованиям задачи и отметить те клетки, которые должны быть закрашены. Также, согласно требованиям задачи, нужно подписать начальную клетку буквой А, а конечную буквой – Б. На рисунке 3 показана готовая обстановка, в которой созданы

нужные стены, введены подписи и отмечены клетки, которые должны быть закрашены.

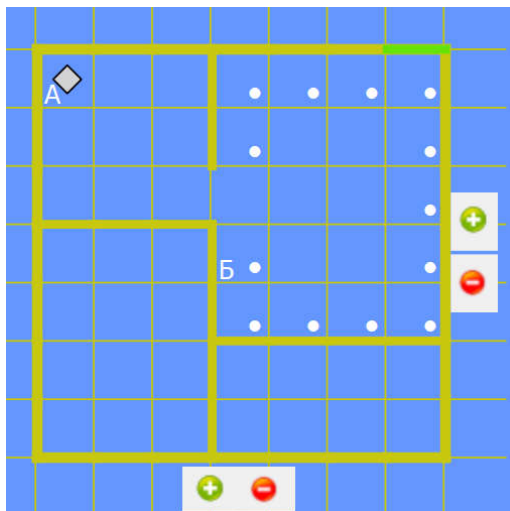


Рисунок 3 – Готовая обстановка в режиме редактирования

Требования к обстановкам для одной задачи:

- количество файлов обстановки для задачи: от 2 до 5, чтобы обеспечить проверку работоспособности алгоритма в различных условиях;
- файлы должны быть сохранены в отдельную директорию;
- имена файлов должны быть уникальными и отражать принадлежность к задаче (например, Задача1_1.fil, Задача1_2.fil).

Также на подготовительном этапе необходимо создать логическую схему практикума, включающую перечень разделов (тем) – крупных блоков, объединяющих задачи по принципу общности тематики или уровня сложности; перечень задач внутри каждого раздела, уникальные идентификаторы (xml:id) для каждого элемента, а также наименования (xml:name) для каждого раздела и каждой задачи, которые будут отображаться в интерфейсе среды «КуМир».

2. Этап написания XML-файла практикума

Итак, подготовительный этап закончен, если созданы каталоги, содержащие задачи и файлы обстановок. После подготовки всех ресурсов можно приступать к созданию главного файла курса, который представляет собой строго структурированный XML-документ, соответствующий внутреннему формату среды «КуМир».

Файл начинается с XML-декларации и содержит корневой элемент <KURS>, внутри которого последовательно размещаются разделы и служебные поля (рис. 4).

```
<?xml version="1.0" encoding="UTF-8"?>
<KURS root="true" xml:id="0" xml:name="Название практикума">
  <!-- Здесь размещаются разделы (темы) -->
  ...
</KURS>
```

Рисунок 4 – Фрагмент XML-документа

- root="true" – обязательный атрибут для корневого элемента.
- xml:id="0" – фиксированный идентификатор корня.
- xml:name – заголовок всего практикума, отображаемый в интерфейсе.

Каждый раздел оформляется как вложенный элемент <T> с атрибутом root="true" (рис. 5).

```
<T root="true" xml:id="1" xml:name="Знакомство">
  <DESC>Решите пробную задачу.</DESC>
  <CS>КуМир</CS>
  <ISP xml:ispname="Robot"/>
  <READY>>false</READY>
  <MARK/>
</T>
```

Рисунок 5 – Обязательные компоненты раздела:

- xml:id – уникальный идентификатор раздела.
- xml:name – название раздела.
- <DESC> – описание раздела.
- <CS> – среда («КуМир»), значение всегда фиксировано.
- <ISP> – указание исполнителя. Для раздела-контейнера тег остаётся пустым, но атрибут xml:ispname="Robot" обязателен.
- <READY> – служебный флаг готовности (false означает, что раздел ещё не пройден).
- <MARK/> – пустой тег для служебных отметок.

При создании задачи (рис. 6) следует учитывать, что задача помещается внутри соответствующего раздела и оформляется элементом <T> с атрибутом root="false".

```
<T root="false" xml:id="101" xml:name="Обход клеток">
  <DESC>Робот должен закрасить все отмеченные клетки и
  перейти в клетку Б.</DESC>
  <CS>Кумир</CS>
  <ISP xml:ispname="Robot">
    <ENV>обстановки\1A-1.fil</ENV>
    <ENV>обстановки\1A-2.fil</ENV>
    <ENV>обстановки\1A-3.fil</ENV>
  </ISP>
  <PROGRAM>задачи\obhod.kum</PROGRAM>
  <READY>>false</READY>
  <MARK/>
</T>
```

Рисунок 6 – Структура задачи

– <ISP> – внутри этого тега перечисляются все подготовленные файлы обстановок. Каждый файл указывается в отдельном теге <ENV>. Путь задаётся относительно местоположения основного XML-файла с использованием обратной косой черты (\).

– <PROGRAM> – путь к файлу-шаблону программы.

В завершении файла после закрытия всех разделов корневой элемент <KURS> должен содержать пустые теги <DESC/> и <MARK/>, после чего он также закрывается.

Приведем некоторые технические замечания, которые необходимо учитывать для корректной работы создаваемого практикума:

– Уникальность идентификаторов: все значения атрибутов xml:id в пределах одного файла должны быть строго уникальными. Нарушение этого требования может привести к некорректной загрузке курса или ошибкам при переходе между задачами.

– Корректность путей к файлам: среда «КуМир» чувствительна к разделителям путей. Рекомендуется использовать обратную косую черту (\), как

в примерах выше. Пути не должны содержать недопустимых символов и должны вести к реально существующим файлам.

– Кодировка символов: файл обязательно должен быть сохранён в кодировке UTF-8 (без BOM). Это обеспечит корректное отображение кириллических символов в названиях и описаниях. Наличие декларации `<?xml version="1.0" encoding="UTF-8"?>` в первой строке обязательно.

– Проверка работоспособности: после создания всех компонентов необходимо выполнить тестовый запуск практикума в среде «КуМир» и убедиться, что все разделы и задачи отображаются в окне навигации, а также при наличии нескольких файлов обстановки убедиться, что среда последовательно предлагает их обучающемуся.

Разработанный по приведенному описанию XML-файл представляет собой полноценный автоматизированный практикум, интегрируемый в среду «КуМир». Процесс его создания включает три последовательных этапа: содержательное проектирование (определение структуры, формулировка заданий), ресурсное наполнение (подготовка файлов обстановок и программ) и техническую реализацию (кодирование XML). Соблюдение всех изложенных требований гарантирует корректную работу практикума, который можно использовать в учебном процессе для автоматизированной проверки правильности решения обучающимися задач по программированию.

Литература

1. Поляков К. Ю. Практикумы с автоматической проверкой решений в среде «КуМир» [Электронный ресурс]. – URL: <https://pubdoc.ru/doc/276855/praktikumy-s-avtomaticheskoy-proverkoj-reshenij-v-srede-kumir> (дата обращения: 08.05.2026).

2. Среда программирования «КуМир» : материалы, ресурсы, методические разработки [Электронный ресурс] / К. Ю. Поляков. – Санкт-Петербург. – URL: <https://kpolyakov.spb.ru/school/kumir.htm> (дата обращения: 08.05.2026).