

**Поминутное планирование**  
**45 минутного занятия для подготовительных групп**  
**дошкольного образовательного учреждения**  
**«Осваиваем программирование за один урок»**  
**А.Г.Кушниренко, А.Г.Леонов, М.В.Райко**  
**Отдел учебной информатики ФГУ ФНЦ НИИСИ РАН**  
**Январь 2016**

**1.а.** Занятие начинается с рассказа о Роботе-Вертуне и принципе программного управления.

**Легенда о Роботе-Вертуне.** В космическом пространстве летают передвижные космодромы. Путешествуя между планетами, космические корабли делают посадки на этих космодромах. Космодромы делают из квадратных плит. Хотя плиты и прочные, но при взлете космического корабля они портятся и их нужно чинить – закрашивать поврежденные места специальной краской.

Учитель может продемонстрировать старт космического корабля на своем планшете, показав первые 30 секунд видеоролика с запуском отечественного космического корабля: <http://www.youtube.com/watch?v=P7BV1Wf285g>

После просмотра ролика нужно обсудить с детьми, почему на ремонт космодрома лучше посылать не человека, а какого-нибудь робота?

**Возможный ответ:** после старта поверхность космодрома очень горячая и человек на ней, даже в специальном скафандре, работать не сможет.

Вертун по своей инициативе ничего не делает. Чтобы Вертун сделал что-то полезное – например ремонт – им нужно командовать, давать ему команду за командой. Командовать Вертуном может человек или компьютер. Чтобы починить большой космодром, нужно будет дать Вертуну много команд и в ходе работы легко запутаться.

**Принцип программного управления.** Чтобы не запутаться, отдавая команды Вертуну, человек может записать план работы Вертуна по ремонту в специальную Книжку или в память компьютера. Такая запись плана называется **Программой**. Когда наступит время делать ремонт – эту программу нужно будет шаг за шагом выполнить. Это выполнение может делать человек, но лучше поручить командование Вертуном по заранее составленной программе компьютеру.

На этом занятии мы научимся составлять программы управления Вертуном, проходя 13 Уровней компьютерной игры.

**1.б.** Вертун живет на клетчатой поверхности, замощенной квадратными плитками. Между некоторыми клетками есть стены. Вертун понимает и умеет выполнять четыре команды:

вперед	закрасить
налево	направо

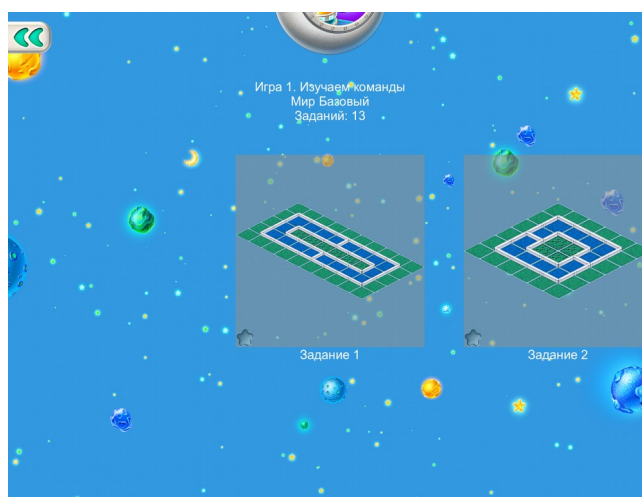
Эти команды будем изображать картинками (пиктограммами):



**Учитель:** «Чтобы вы хорошо запомнили команды Вертуна, даю каждому из вас памятку с этими командами. Эта памятка магнитная. Вы можете повесить ее дома на холодильник и рассказать родителям про Вертуна.»

Детям выдается памятка с командами Вертуна – магнит на холодильник  
Вообще-то Вертун – робот космический. Он предназначен для ремонта космических платформ-космодромов. Космические корабли взлетают и садятся на космодром, при этом некоторые плиты повреждаются и их нужно чинить – закрашивать специальной краской.

**1.в.** Знакомство с Игрой. Учитель, используя проектор, демонстрирует детям на учительском планшете как запустить ПиктоМир и начать игру .



- **Материал для учителя. Терминология.** В учебной программной системе Пиктомир **программой** называют план управления Роботом, записанный по определенным правилам. Программа в ПиктоМире состоит из частей, называемых **алгоритмами**. В программе всегда есть один главный алгоритм, с него начинается выполнение программы. В программе могут быть и другие алгоритмы, они имеют однобуквенные имена: А, Б, В, ... и называются вспомогательными алгоритмами.
- В учебной программной системе КуМир используется та же терминология: программа состоит из алгоритмов, есть главный алгоритм и вспомогательные алгоритмы.
- В других программных системах, учебных и производственных, также используется термин программа, а вместо термина алгоритм используются термины подпрограмма и/или функция.
- В ПиктоМире и Кумире , как и в других системах разработки программ, есть два режима выполнения программы: непрерывный и пошаговый.

- Программное обеспечение занятия состоит из одной Игры. Игра состоит из 13 Уровней. На каждом уровне ребенку предлагается задание. Для прохождения уровня нужно составить программу, выполняя команды которой, Вертун выполняет задание. Шаблон программы также дается в задании и не может быть изменен.

### 1.г. Дети открывают уровень 1 .



Учитель показывает детям:

- Как добавить пиктограмму в программу.  
Первый способ: «хлопнуть» на пиктограмме (она начнет подпрыгивать), затем «хлопнуть» в нужном месте программы.  
Второй способ: перетащить пиктограмму в нужное место программы.  
Третий способ: найти нужную пиктограмму в программе и перетащить ее копию в нужное место.
- Как удалить пиктограмму из алгоритма.  
Для удаления нужно перетащить пиктограмму за рамку – границу алгоритма.

Учитель показывает детям, как он заполняет шаблон для программы и как выполняет готовую программу.



1.д. Дети решают самостоятельно задание 2.



Шаблон программы:

*Главный Алгоритм*


Решение:

*Главный Алгоритм*


1.е. Учитель: «Откройте задание уровня 3.»



**Указание для учителя.** На уровне 3 программа получается длинная. Составлять длинные программы помогает «КОПИЛКА КОМАНД ПиктоМира». Работу копилки объясним на примере решения задачи 3. Учитель использует проектор и показывает на экране как работает копилка.

Открываем копилку: нажимаем маленькую кнопку с бирюзовой стрелкой.



После этого начинаем давать команды Вертуну, нажимая на пиктограммы команд в нужном порядке. Мы видим, как после каждого нажатия Вертун выполняет команду, а пиктограмма команды летит в копилку.



Как работать с копилкой:

- Если последняя попавшая в копилку команда оказалась неправильной, то ее можно отменить. Для этого нужно «кликнуть» на пиктограмме этой команды в копилке. Пиктограмма при этом исчезнет, а Вертун вернется в то состояние, в котором он был до выполнения отмененной команды.
- Выбросить все команды из копилки можно нажав на красную кнопку-корзину. Вертун вернется в исходное положение.

Когда Вертун выполнил свою работу, все нужные команды в порядке снизу вверх сложены в копилке, при этом видны только последние команды:

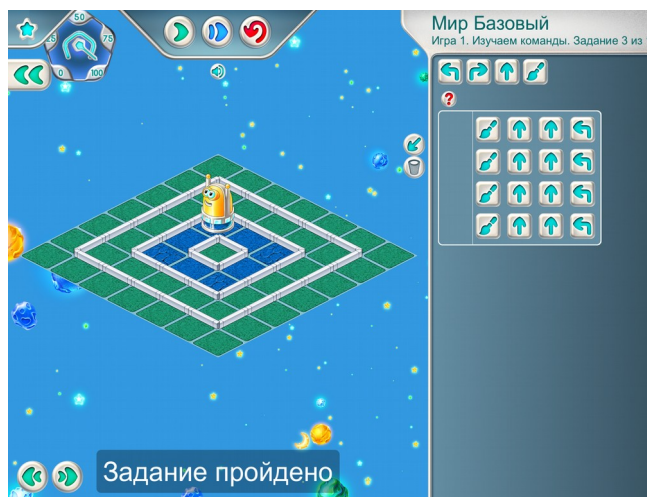


Чтобы перенести эти команды в шаблон для программы, нажмите стрелочку справа от шаблона программы. Получится правильная программа в нужном месте:



**1.ж. Учитель:** «Давайте переделаем программу так, чтобы Вертун вернулся в исходную точку в том же положении.»

Учитель заполняет программу недостающими пиктограммами. И выполняет ее.



**Учитель:** «В этой программе 4 одинаковые строчки. Поэтому программу можно сделать короче, зашифровав повторяющуюся строчку буквой А. Перейдем на следующий уровень.»



На экране видно, что строчка из 4 команд зашифрована буквой А. Чтобы выполнить эту строчку 4 раза, вставьте в программу 4 буквы А



**Учитель:** «Проверим зашифрованную программу. Нажимайте синюю кнопку, пока программа не будет выполнена до конца.»

**1.з. Учитель:** «Есть и другой способ сократить программу. Перейдем на уровень 5.



**Учитель:** «Белые кружочки — называются повторителями. Если вставить в программу повторитель с четырьмя точками, то строчка после повторителя будет выполнена 4 раза. Заполним строчку после повторителя и посмотрим как работает программа.»

**Учитель:** «Программы в заданиях 3, 4 и 5 делают одну и ту же работу. Программа в задании 3 составлена из 16 пиктограмм. Программа в задании 4 — из 9, а программа в задании 5 составлена всего лишь из 5 пиктограмм.»


**1.и.** Дети самостоятельно выполняют задание уровня 6





Шаблон программы:

Главный Алгоритм



Алгоритм 








Решение

Главный Алгоритм






Алгоритм 


		
		

1.к. Дети самостоятельно выполняют задание уровня 7










Шаблон программы:

Главный Алгоритм

Решение:

Главный Алгоритм

1.л. Учитель: «Откройте уровень 8.



Учитель рассказывает детям как составить программу для этого задания.

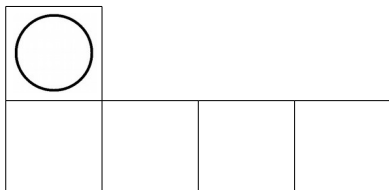
Учитель: «Для ремонта одной стороны квадрата нужно дать Роботу команду закрасить и 7 команд вперед :



Шесть команд вперед запакуем в *Алгоритм А* с помощью повторителя 6. А седьмую команду вставим в *Главный Алгоритм*. После семи шагов вперед дадим Роботу команду налево. Робот должен пройти 4 стороны квадрата, поэтому в *Главный Алгоритм* ставим повторитель 4.»

Шаблон программы:

*Главный Алгоритм*

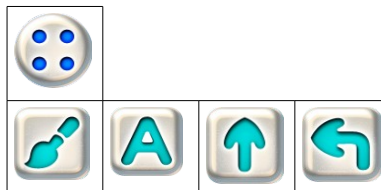


*Алгоритм*

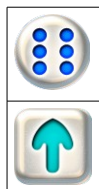


Решение:

*Главный Алгоритм*



*Алгоритм*



**Замечание для учителя:** шаблон задания демонстрирует детям важный новый прием: использование повторителя внутри вспомогательного алгоритма. Этот прием будет использоваться в дальнейшем.

Учитель: «Составьте программу и выполните ее.»

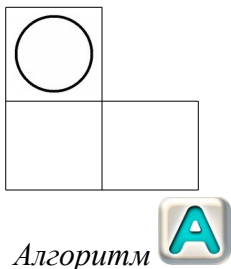
11. 1.м. Учитель предлагает детям самостоятельно выполнить задания на уровнях 9, 10 и

Уровень 9.

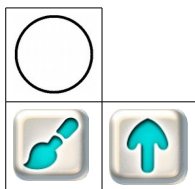


Шаблон программы:

Главный Алгоритм

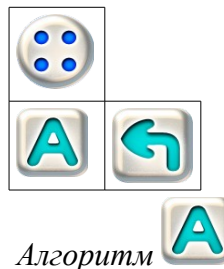


Алгоритм

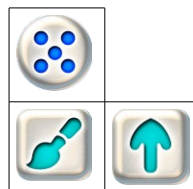


Решение:

Главный Алгоритм



Алгоритм



# Уровень 10.



Шаблон программы:

*Главный Алгоритм*



*Алгоритм* **A**


Решение

*Главный Алгоритм*



*Алгоритм* **A**


# Уровень 11.



Шаблон программы:

*Главный Алгоритм*


*Алгоритм*


Решение

*Главный Алгоритм*


*Алгоритм*


1.н. Учитель: «Перейдем на Уровень 12.»



Учитель рассказывает детям как составить программу для этого задания.

Учитель: «Воспользуемся копилкой»



Заметим, что таких уголков 3, отправим пиктограммы из копилки в Алгоритм А.



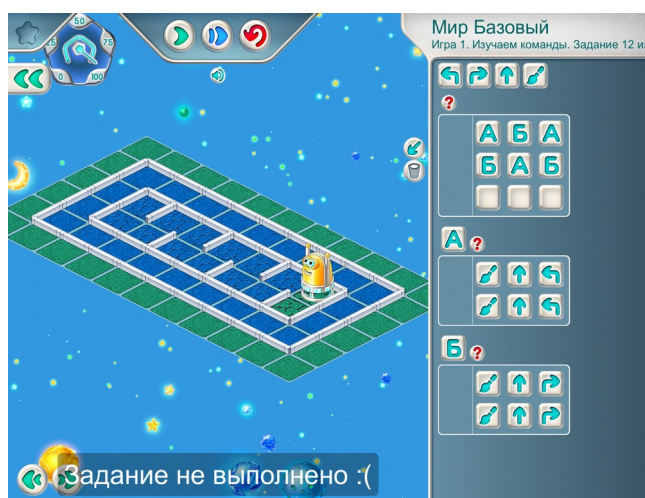
Следующий уголок с поворотом направо. И их тоже 3 . Поэтому заполним *Алгоритм Б* пиктограммами для прохода второго уголка.



Для заполнения *Главного Алгоритма* можно воспользоваться копилкой:



Отправим их в шаблон программы:



Дополните недостающие пиктограммы и проверьте работу программы.»



1.о. Учитель предлагает детям самостоятельно играть на уровне 13.





Шаблон программы:





Главный Алгоритм


Алгоритм **A**







Алгоритм **Б**


Решение:

Главный Алгоритм

	<b>A</b>
<b>A</b>	<b>Б</b>
	
<b>Б</b>	<b>A</b>
<b>A</b>	

Алгоритм **A**

Алгоритм **Б**

