**Исполнитель работы:** Пиляк Ксения Юрьевна.

**Название работы:** «Алгоритмический игровой исполнитель».

**Научный руководитель:** Сергиевич Николай Владимирович, кандидат физико-математических наук, доцент.

**Основные цели и актуальность работы:** Раздел «Основы алгоритмизации и программирования» обычно вызывает трудности у большинства учащихся. Во-первых, это связано с еще недостаточным уровнем сформированности алгоритмического мышления. Во-вторых, часто случается так, что учащиеся не до конца понимают, для чего нужно программирование и как алгоритмизация связана с разработкой программ, которыми они пользуются ежедневно. В-третьих, учащиеся могут быть и не заинтересованы учиться программированию, поскольку многие задачи из школьных учебников ограничиваются вводом и выводом на экран какой-то информации и не носят творческого характера, из-за чего дети теряют интерес к предмету. Для обучения детей программированию и алгоритмизации в разное время были созданы алгоритмические исполнители такие как «Чертежник», «Кенгуренок», «Черепашка» и другие. Они намного легче языков программирования высокого уровня, а также позволяют выполнять интересные задания. Но эти исполнители, как правило, подходят для подготовленных учащихся, уже изучивших принципы алгоритмизации.

Конечно, ничто не заинтересует ребенка так, как игра. Поэтому целью данной курсовой работы является написание алгоритмического игрового исполнителя для использования учащимися на уроках информатики. Изучая алгоритмизацию в игровой форме, ученики могут развить мышление, изучить основные виды алгоритмов, а также научиться составлять их.

**Практическая реализация и внедрение результатов:** Алгоритмический игровой исполнитель «Робот» – это образовательная программа, с помощью которой осуществляется обучение учащихся основам алгоритмизации. Вся программа состоит из пяти разделов: «Линейные алгоритмы», «Алгоритмы с ветвлением», «Циклические алгоритмы», «Подпрограммы», «Контрольные задания». Каждый раздел сопровождается небольшим теоретическим материалом, в котором приводятся определение каждого типа алгоритмов, их блок-схемы, особенности использования. Также приводятся программные блоки алгоритмов на языке программирования Pascal. После теоретического материала пользователю предлагается выполнить четыре задания различных уровней сложности. В разделе «Контрольные задания» приводится сводная таблица по всем изученным видам алгоритмов и предлагается выполнить задания по одному на каждую из изученных тем. Доступ к заданиям и разделам не ограничен, таким образом, пользователь в любой момент может выбрать интересующую его тему или пройти ее заново.

На рабочем поле размером 6х6 находится исполнитель Робот (рис. 1.2). Некоторые клетки на поле закрашены желтым цветом. Это означает, что Роботу для достижения цели необходимо пройти все желтые клетки. В заданиях по теме «Алгоритмы с ветвлением» есть коричневые клетки – препятствия, в которые Робот попасть не может, их нужно перепрыгивать. Справаот поля находятся кнопки, которые предназначены для управления поведением Робота на поле. Робот имеет возможность переходить на одну клетку вниз, вправо, вверх, влево. В алгоритмах с ветвлением для того, чтобы перепрыгнуть препятствие, необходимо сделать ветвление if is wall (кнопка «**is wall**»), после чего появится возможность использовать кнопки для перепрыгивания Робота через препятствие – они становятся активными. В условных алгоритмах в выпадающем списке можно выбирать количество повторений в цикле for. При нажатии кнопок соответствующие им команды появляются в программном блоке и в блок-схеме, а в случае вспомогательных алгоритмов вызов процедуры появляется в основной программе, а другие действия – в процедуре.

При этом Робот пока не совершает никаких действий на поле. Чтобы привести Робота в движение, необходимо нажать кнопку «**Пуск**». Чтобы отменить последнюю неверную команду, нужно нажать кнопку «**Отмена**». Кнопка «**В начало**» предназначена для возвращения Робота в исходное положение. Для получения справки по использованию программы имеется кнопка «**Помощь**». Пользователь имеет возможность использовать лишь ограниченное количество команд, что позволяет научиться составлять более эффективные алгоритмы. Внизу экрана есть строка, в которой выводятся комментарии. Если в ходе выполнения алгоритма Робот выходит за пределы поля, то выводится соответствующее сообщение об ошибке, где указывается, в какой строке была вызвана неверная команда. Если пользователь успешно справляется с заданием, то выводится надпись «Молодец!».

Программа написана на языке программирования Java. Программу можно разделить на две большие части: область теоретического обучения (класс Title) и область практического обучения (класс Window). В области теоретического обучения приводятся основные теоретические сведения по каждому из разделов. В области для практического обучения представлен пользовательский интерфейс для выполнения заданий. Его представляют следующие компоненты: рабочее поле (класс Field), исполнитель Робот (класс Robot), программный блок (класс ProgramField), блок-схема (класс Blocks), кнопки управления исполнителем (класс ControlButton).

Интерфейс TypeOfCommands содержит строковые константы – команды действий, устанавливаемых кнопкам, используемых в программе.

На основе созданной программы была составлена UML-диаграмма классов.

**Основные результаты, полученные в работе, выводы, перспективы продолжения работы**: Алгоритмический игровой исполнитель «Робот» предназначен для обучения основам алгоритмизации детей младшего и среднего школьного возрастов. Может применяться учителем в качестве дополнительного или обучающего материала на уроках информатики, кружках и факультативах при изучении темы «Основы алгоритмизации и программирования». В ходе курсовой работы приобретены навыки программирования на языке Java, исследованы основные понятия темы «Алгоритмизация», изучен язык UML. В дальнейшем планируется совершенствовать программу, добавлять в нее новые возможности.