

4. Стельмашук, Н.Т. Метод формальных производных для решения задачи Коши для одной системы дифференциальных уравнений в частных производных / Н.Т. Стельмашук, В.А. Шилинец // Дифференциальные уравнения. – 1993. – № 11. – Т. 29. – С. 2019–2020.
5. Stelmashuk, N.T. The solution of the boundary value problem for a system of equations in formal derivatives by means dual differential operators / N.T. Stelmashuk, V.A. Shylinets // Труды института математики НАН Беларуси. – 2004. – № 2. – Т. 12. – С. 170–171.
6. Стельмашук, Н.Т. О преобразовании к каноническому виду системы линейных уравнений в частных производных с помощью двойных дифференциальных операторов / Н.Т. Стельмашук, В.А. Шилинец // Весці НАН Беларусі. Сер. фіз.-мат. навук. – 2008. – № 2. – С. 61–65.
7. Стельмашук, Н.Т. Об одном исследовании системы Максвелла с помощью F-моногонных функций / Н.Т. Стельмашук // Журнал вычислительной математики и математической физики. – 1967. – № 2. – Т. 7. – С. 431–436.
8. Стэльмашук, М.Т., Шылінец У.А. Пабудова інтэгральных выяўленняў для функцыянальна-інварыянтных рашэнняў сістэмы дыферэнцыяльных раўнанняў Максвэла / М.Т. Стэльмашук, У.А. Шылінец // Весці БДПУ. – 1999. – № 2. – С. 147–150.
9. Гусев, В.А. об одном обобщении ареолярных производных / В.А. Гусев // Bul. stiint. al Institut. politehnic Timisoara. – 1962. – F. 2. – T. 7. – P. 223–238.

## ИСПОЛЬЗОВАНИЕ РАЗНОУРОВНЕВЫХ ЗАДАЧ В ОБУЧЕНИИ SCRATCH-ПРОГРАММИРОВАНИЮ

**Бабаева А.Б., Ораева О.О.,**

*студентки 2-го курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь*

*Научный руководитель – Булгакова Н.В., старший преподаватель*

Обучение программированию в современной школе считается довольно трудным разделом информатики. Для того, чтобы освоить программирование, ученики должны владеть логическим мышлением и иметь математическую подготовку. Использование в процессе обучения специализированных сред, например, Scratch, позволит сделать процесс обучения школьников программированию интересным и увлекательным занятием. В связи с этим актуальной является задача рассмотрения использования разноуровневых заданий в обучении Scratch-программированию. Использование разноуровневых заданий позволяет решить целый ряд педагогических задач и может быть рассмотрено как дидактический инструмент учителя, использование которого ускоряет и углубляет понимание учащимися структуры знаний, а также способствует поэтапному формированию практических навыков программирования.

Среда программирования Scratch представляет собой визуальную объектно-ориентированную среду, которая имеет простой и удобный интерфейс. Используя эту среду можно не только программировать, но и реализовывать графику и моделирование.

*Целью настоящей работы* является рассмотрение возможностей использования разноуровневых задач при обучении школьников Scratch-программированию.

**Материал и методы.** Материал исследования – Scratch как язык и среда программирования; дидактические возможности использования в обучении программированию разноуровневых задач. В работе используются методы исследования экспериментально-теоретического уровня: анализ и синтез, изучение и обобщение, формализация.

**Результаты и их обсуждение.** Среда программирования Scratch представляет собой средство визуального обучения программированию школьников младших и средних классов. Визуальное представление информации повышает степень вовлеченности учащегося в процесс получения знаний, визуальная информация быстрее воспринимается и легче запоминается. Особую значимость в обучении среда Scratch приобретает благодаря возможности создавать анимацию.

Как известно из когнитивной психологии, восприятие и обработка новой информации в процессе изучения нового материала соотносятся с понятиями и способами действий, известными ученику. Новая информация, поступающая в мозг, структурируется, образуя в сознании концептуальные сети, при этом устанавливаются связи между известными понятиями и новыми знаниями [1]. В этой связи особого внимания заслуживает порядок решения задачи с использованием компьютера. Он включает в себя следующие этапы:

- содержательная постановка задачи (на ее основе разрабатывается формализованная модель предметной области, объекта);

- разработка алгоритма (последовательность команд, приводящая к решению задачи);
- составление программы (собственно программирование, которое в узком смысле понимается как кодирование алгоритма решения задачи, а в широком смысле включает и содержательную постановку задачи).

На этапе содержательной постановки задачи ученик должен собрать и структурировать все сведения, относящиеся к данной задаче, указать все входные и выходные параметры, рассмотреть способы их преобразования. Вообще говоря, содержательная постановка задачи – творческий процесс, одну и ту же задачу разные люди могут сформулировать по-разному. Но от содержательной постановки задачи во многом зависит успешность программирования в целом.

Сложность программирования растет по мере увеличения сложности поставленной задачи. На этапе формализации поставленную задачу необходимо разбить на несколько взаимосвязанных подзадач, которые можно программировать по отдельности. На этом же этапе выбираются способы некоторых вычислений, строится математическая модель объекта.

Математическая модель должна удовлетворять ряду требований: полно (адекватно) описывать основные закономерности функционирования объекта; ориентироваться на решение некоторого класса задач и использовать определенные методы их решения; ориентироваться на применение имеющихся вычислительных средств.

При разработке конкретного алгоритма решения задачи нам, фактически, необходимо составить два алгоритма: алгоритм потока информации (прием, размещение, сортировка и переработка данных) и алгоритм решения математической задачи. Эти два алгоритма взаимосвязаны: с одной стороны, при выборе алгоритма решения математической задачи учитывается возможность обеспечить его данными, но, с другой стороны, содержание информационных массивов, форм и методов хранения данных зависят от алгоритма решения. Этап разработки алгоритма решения задачи завершается созданием подробной блок-схемы алгоритма.

Последний этап решения задачи – собственно программирование: производится кодировка (запись алгоритма на языке программирования) и отладка программы (поиск и исправление ошибок в программе).

Обучение программированию сводится к тому, что ученик должен научиться самостоятельно выполнять все указанные этапы решения задач на ЭВМ. Обучение решению задач в среде Scratch может быть рассмотрено на нескольких уровнях: репродуктивном (воспроизвести образец, для предъявления которого требуется визуализация); продуктивном (ученикам предъявляется визуальный образ результата, методы и средства решения задачи они подбирают самостоятельно); творческом (ученикам предъявляется только направление исследования, тема).

Для эффективного решения задач творческого уровня ученикам необходимо овладеть техниками решения задач на репродуктивном и продуктивном уровнях. Использование разноуровневых заданий в обучении Scratch-программированию способствует формированию фундаментальных знаний по информатике, развитию умений и навыков составления алгоритмов и их реализации компьютерными средствами [2].

**Заключение.** Использование разноуровневых задач в обучении программированию позволяет решить целый ряд педагогических задач:

- реализует дидактический принцип движения по материалу от простого к сложному;
- способствует более полному и успешному освоению основных навыков программирования за счет вовлечения учеников в практическую деятельность по решению задач;
- облегчает процесс обучения вследствие интеграции полученных знаний в структуру уже имеющихся, стимулирует логическое и алгоритмическое мышление учащихся;
- развивает креативность, творческий потенциал школьников.

Среда программирования Scratch обладает всеми необходимыми средствами для решения этих задач.

1. Баженова, И.В. Визуализация знания как метод когнитивного подхода к обучению программированию // Информационно-образовательная среда вуза (Совместно с КГПУ имени В.П. Астафьева). Решетневские чтения. – 2014. – С. 281–284.
2. Корнилов, В.С. Использование среды программирования Scratch в преподавании школьного курса информатики в классах коррекционно-компенсирующего обучения / В.С. Корнилов, В.А. Зарянкин // Вестник РУДН, Сер. Информатизация образования. – 2014. – № 1. – С. 51–54.

## РАЗРАБОТКА ПРОГРАММЫ “LIFE SIMULATION”

*Бачун Р.В.,*

*учащийся 2-го курса Оршанского колледжа ВГУ имени П.М. Машерова,  
г. Орша, Республика Беларусь*

*Научный руководитель – Алейников М.А., магистр педагогических наук*

Клеточный автомат – дискретная модель, изучаемая в математике, теории вычислимости, физике, теоретической биологии и микромеханике. Включает регулярную решётку ячеек, каждая из которых может находиться в одном из конечного множества состояний, таких как 1 и 0. Решетка может быть любой размерности. Для каждой ячейки определено множество ячеек, называемых окрестностью. К примеру, окрестность может быть определена как все ячейки на расстоянии не более 2 от текущей (окрестность фон Неймана ранга 2). Для работы клеточного автомата требуется задание начального состояния всех ячеек и правил перехода ячеек из одного состояния в другое. На каждой итерации, используя правила перехода и состояния соседних ячеек, определяется новое состояние каждой ячейки. Обычно правила перехода одинаковы для всех ячеек и применяются сразу ко всей решётке [1; 3].

*Цель исследования* – создать приложение и реализовать в нем алгоритм для понимания происходящих в мире процессов, рассматривания их с точки зрения симуляции, а также исследования различного рода случайных явлений и выявления их закономерностей.

**Материал и методы.** В качестве среды разработки для реализации симуляции на языке программирования C# был выбран движок Unity. Для изучения движка была использована книга “Unity в действии. Мультиплатформенная разработка на C#.” За основу для реализации алгоритмов был выбран язык программирования C#.

**Результаты и их обсуждение.** Программа “Life Simulation” – это простейшая симуляция жизни, созданная для исследования поведения клеток.

Всего в симуляции программы “Life Simulation” 4 точки: зеленая – клетка, желтая – еда, красная – погибающая клетка и невидимая – создана для генерации случайного пути клетки.

Клетка имеет несколько функций – деление и поиск еды. Если клетка не нашла еду в течении минуты – она погибает и превращается в красную точку, живые клетки могут ее съесть.

Если клетка нашла еду, время жизни увеличивается на минуту, за одну единицу еды, клетка увеличивается на 10%, если клетка съела 10 единиц еды – она делится.

Еда создается каждые 10 секунд в случайном месте, если в течении 60 секунд еда не была съедена – она уничтожается.

Движение клетки случайно. Всего 4 позиции (вперед, назад, лево, право). Создается случайное число от 1 до 4, каждая цифра соответствует определенной позиции. После генерации клетка движется в нужном направлении на 1, когда клетка доходит до позиции алгоритм сбрасывается снова. Результат работы алгоритма можно наблюдать на рисунке 1. На нем изображена работа клеточного автомата – а именно взаимодействие клеток, алгоритм взаимодействия которых описан выше.