

5. Лазаренко, О. И. Диагностика и коррекция выразительности речи детей : методическое пособие / О. И. Лазаренко. – Москва : Сфера, 2009. – 64 с.
6. Современное образование : проблемы и перспективы развития : монография / Т. А. Ларина, Т. И. Грицкевич, А. Н. Захарова [и др.]. – Чебоксары : Среда, 2019. – 72 с.
7. Лопатина, Л. В. Логопедическая работа по развитию интонационной выразительности речи дошкольников : учебное пособие / Л. В. Лопатина, Л. А. Позднякова. – Санкт-Петербург : СОЮЗ, 2006. – 151 с.
8. Пономарева, В. А. Исследования просодической стороны речи у дошкольников с общим недоразвитием речи / В. А. Пономарева // Научно-методический электронный журнал Концепт. – 2016. – № 13. – С. 51–55. – URL: <http://e-koncept.ru/2016/56171.htm> (дата обращения: 02.05.2023).
9. Фомичева, М. Ф. Воспитание у детей правильного звукопроизношения : практикум по логопедии : учебное пособие для учащихся педагогических училищ по специальности 03.08 «Дошкольное воспитание» / М. Ф. Фомичева. – Москва : Просвещение, 1989. – 239 с.
10. Хватцев, М. Е. Логопедия. Теория и практика : учебное пособие. / М. Е. Хватцев. – 3-е издание, исправ. – Санкт-Петербург : Наука и Техника, 2022. – 320 с.
11. Чирвина, О. В. Диагностика интонационной стороны речи по методике Лазаренко О. И. / О. В. Чирвина // Наша сеть : образовательная социальная сеть. – URL: <https://nsportal.ru/detskiy-sad/razvitie-rechi/2015/02/15/diagnostika-intonatsionnoy-storony-rechi-po-metodike-lazarenko> (дата обращения: 02.05.2023).
12. Шик, Н. С. Диагностика состояния просодических компонентов речи у дошкольников с ФФНР и нарушениями зрения / Н. С. Шик // Сборник диагностических методик. – [сайт]. – URL: <http://www.psmethodiki.ru/index.php/doshkol/dproch/131-diagnostika-sostoyaniya-prosodicheskikh-komponentov-rechi-u-doshkolnikov-s-ffnr-i-narusheniyami-zreniya> (дата обращения: 04.05.2023).

*Гарколь Н.С., канд. техн. наук, доцент кафедры теоретических основ информатики,
Пономаренко О.П., канд. филос. наук, доцент кафедры философии и культурологии
Алтайский государственный педагогический университет
г. Барнаул*

МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ СОВРЕМЕННЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ ПРИ ФОРМИРОВАНИИ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ

Аннотация. В статье рассмотрено моделирование как метод современных образовательных технологий познания окружающего мира. Целью исследования является целостное рассмотрение, выявление, теоретическое обоснование и апробирование методики применения компьютерных технологий как способа формирования алгоритмического мышления студентов педагогического вуза. Определены платформы образовательных технологий для моделирования движения объектов. Приведены примеры и этапы выполнения заданий, показывающие практическое применение компьютерного моделирования на занятиях.

Ключевые слова: моделирование движения объектов, алгоритмическое мышление, информационные технологии

**N.S. Garkol,
O.P. Ponomarenko**

MODELING THE MOTION OF OBJECTS WITH THE USE OF MODERN EDUCATIONAL TECHNOLOGIES FOR THE DEVELOPMENT OF ALGORITHMIC THINKING

Abstract. *This article discusses modeling as a method of using modern educational technologies for the study of the world around us. The aim of the research is to comprehensively consider, identify, theoretically justify, and test the methodology of using computer technologies as a way to develop algorithmic thinking among students of a pedagogical university. We have identified educational technology platforms for modeling the motion of objects. Examples and steps for completing tasks are provided, demonstrating the practical use of computer modeling in the classroom.*

Keywords: modeling the motion of objects, algorithmic thinking, information technologies

Современные образовательные технологии открывают широкие возможности для реализации методов моделирования на уроках информатики. Моделирование движения объектов – это очень увлекательное и познавательное занятие, помогающее учащимся лучше понять физические законы и принципы работы различных систем. Изучение основ компьютерного моделирования знакомит будущих педагогов с современными информационными технологиями, которые в дальнейшем могут быть применены на уроках информатики. Актуальность исследования определяется важностью использования компьютерных технологий для развития логического и алгоритмического мышления будущего учителя. В практической работе это будет способствовать развитию алгоритмического мышления учащихся.

Научная новизна исследования заключается в теоретическом обосновании метода моделирования в процессе формирования алгоритмического мышления обучающихся. Развитие мыслительной деятельности будущего педагога способствует последовательности и логичности действий в разрешении педагогических ситуаций, решению теоретических и практических задач посредством последовательности действий.

В научной литературе алгоритмическое мышление рассматривается как методологическая составляющая компетенции педагога. Данный вид мышления определяется способностью осуществлять выбор оптимального маршрута достижения образовательной цели (С.О. Алтухова, Н.Ф. Долганова, С.И. Мараджабов, И.Н. Смирнова), составлять и решать задачи (О.В. Чебурина), действовать по заранее установленному правилу или образцу (Е.А. Утюмова).

В изучении основ компьютерного моделирования необходимо не только опираться на теоретическое изучение предмета, но и уделять внимание практической составляющей. Такой процесс моделирования становится особенно интересным для учеников, когда на уроках информатики используют инструменты 3D-моделирования, анимацию, симуляции, что делает процесс обучения более интерактивным и увлекательным. Подобные информационные системы позволяют ученикам интерактивно взаимодействовать с моделями, помогают им глубже понять физические законы, алгоритмы и принципы работы различных систем. В обучение необходимо включать и работу с программным обеспечением, что позволяет развивать навыки программирования, алгоритмического мышления и решения задач.

В качестве современных платформ образовательных технологий для моделирования движения объектов можно использовать следующие:

- Визуальные языки программирования (Scratch, Blockly);
- 3D-моделирование (Blender, Tinkercad);
- Программные среды для физических симуляций (Algodoo, PhET);
- Игровые движки (Unity, Unreal Engine).

Языки программирования Scratch, Blockly позволяют создавать анимации и модели движения объектов с помощью блоков кода, доступных даже для начинающих программистов.

Платформы Blender, Tinkercad обеспечивают создание трехмерных моделей объектов, которые можно анимировать и использовать для различных симуляций.

Программные среды Algodoo, PhET позволяют моделировать движение объектов, учитывая физические законы, такие, как гравитация, трение, импульс.

Unity, Unreal Engine дают возможность создавать интерактивные игры и симуляции с реалистичным движением объектов.

Информационные технологии рекомендуется использовать при изучении дисциплин: физика, химия, астрономия, история, иностранные языки.

Например, можно предложить в качестве проектных тем следующие:

в физике: изучение законов Ньютона, траекторий движения, гравитации;

в разделах механики: моделирование движения машин, ракет, самолетов, механизмов, роботов, а можно предложить создание моделей летательных аппаратов и изучение их движения в воздухе;

в разделах гидродинамики: изучение движения объектов в воде, изучение волн, течений.

в астрономии: моделирование движения планет, звезд, галактик.

в химии: моделирование протекания химических процессов.

в иностранных языках: моделирование виртуальных ситуаций, где можно практиковать разговорные навыки, например, заказывая еду в ресторане, покупая билеты на поезд или общаясь с врачом.

в истории: моделирование исторических событий, создание виртуальных экскурсий по музеям, или посещение исторических мест в виртуальной реальности, изучая культуру и практикуя иностранный язык в контексте. Построение моделей, создание анимационных фильмов или видеороликов способствует пониманию и объективной оценке происходящих событий и может использоваться при изучении любого предмета.

Наиболее распространенным языком программирования является Python, который возможно начинать изучать с 5–8 классов.

Для закрепления навыков программирования можно предложить следующие примерные задания для учеников:

- Создание модели движения автомобиля с учетом сил трения и гравитации.
- Разработка игры, в которой персонаж прыгает и преодолевает препятствия.
- Моделирование движения мяча в игре в баскетбол (футбол, баскетбол, атомов в молекулах).
- Проектирование и создание модели машины (самолета, любого робота), способного перемещаться по заданной траектории.
- Конечно, выбирать задания нужно в соответствии с возрастом и уровнем подготовки учеников, помогая им при этом в решении задач с реальными примерами, а главное – обеспечить им доступ к необходимому программному обеспечению, ресурсам и инструкциям.

На *первом* этапе изучения технологий компьютерного моделирования достаточно выполнить моделирование простых фигур:

1. Создать модель куба, пирамиды или цилиндра с помощью программы 3D-моделирования.
2. Изменить цвет, размер и текстуру модели.
3. Сделать анимацию вращения или перемещения модели.

На *втором* этапе выполнить моделирование движения:

4. Создать модель движущегося объекта (шарик, автомобиль, самолет, робот) с помощью программы визуального программирования (Scratch, Blockly).
5. Добавить анимацию движения, изменения направления, взаимодействия с другими объектами.
6. Создать 3D-модель реального объекта (например, автомобиль, самолет, мяч, здание, мебель и т.д.) с помощью программы 3D-моделирования.

7. Добавить текстуры, материалы и детали, чтобы сделать модель более реалистичной.

8. Анимировать модель, чтобы показать ее функциональность.

На *третьем* этапе выполнить простую симуляцию:

9. Создать модель падения мяча с учетом гравитации с помощью программы физических симуляций (Algodoo, PhET, или функций CSS HTML).

10. Изменить высоту падения, массу мяча и силу гравитации, чтобы наблюдать за изменениями.

11. Создать модель движения маятника или колебания пружины с помощью программы физических симуляций.

12. Изменить параметры модели (длину маятника, жесткость пружины) и наблюдать за изменениями в его движении.

Представленные этапы средств и методов обучения будут максимально влиять на формирование алгоритмического мышления будущих педагогов.

В качестве примера можно привести перемещение «автомобиля» по экрану монитора. Ниже представлен «рабочий алгоритм», который можно использовать как базовый на уроках информатики. Алгоритм позволяет перемещать автомобиль по экрану, реализованный на языке Python. В современном мире программирования язык Python стал одним из самых популярных языков среди начинающих разработчиков и профессионалов благодаря простоте, читаемости и мощным возможностям.

В алгоритме использованы функции библиотеки Pygame, в которой представлен набор программных средств, помогающих разрабатывать любые компьютерные модели, а также создавать двумерные компьютерные игры.

```
import pygame, math, sys
from pygame.locals import *

# Set the screen
screen = pygame.display.set_mode((800*3/2, 600*3/2))
WHITE = (255, 255, 255)
screen.fill(WHITE)

# Initialize the clock
clock = pygame.time.Clock()
fps = 60
class car:
    MAX_FORWARD_SPEED = 2.5
    MAX_REVERSE_SPEED = 2.5

    def __init__(self, car_filename, X, Y, angle):
        #pygame.sprite.Sprite.__init__(self)

        # Load the car image
        self.car = pygame.image.load(car_filename).convert()
        self.car = pygame.transform.rotate(self.car, 180)
        self.car.set_colorkey((255, 255, 255))
        self.car = pygame.transform.scale(self.car, (100, 100))
        # Speed is between 0-3
        self.speed = 0
        # Direction: 0 = right, 1 = down, 2 = left, 3 = up
        self.direction = 1
        # Keep track of rectangle's position
```

```

self.xpos = X
self.ypos = Y
self.angle = angle
# Initialize the car by blitting it
self.display_car()

# EFFECTS: Function to display the car image
def display_car(self):
    self.xpos = self.xpos - self.speed*math.cos(math.radians(self.angle))
    self.ypos = self.ypos + self.speed*math.sin(math.radians(self.angle))
    screen.fill((0, 0, 0))
    screen.blit(pygame.transform.rotate(self.car, self.angle), (self.xpos, self.ypos))
    pygame.display.flip()

# Makes car turn left
def left(self):
    self.angle+=0.5
    #self.direction = (self.direction + 3) % 4
    # for i in range(91):
    #     clock.tick(45)
    #rotated_car = pygame.transform.rotate(self.car, self.angle)
    self.display_car()

# Makes car turn right
def right(self):
    self.angle-=0.5
    #self.direction = (self.direction + 1) % 4
    # for i in range(91):
    #     clock.tick(45)
    #rotated_car = pygame.transform.rotate(self.car, self.angle)
    self.display_car()

def accelerate(self):
    self.speed += 0.3
    # for i in range(90):
    #     clock.tick(30)

    # self.speed = 0

def decelerate(self):
    self.speed -= 0.3
    # for i in range(100):
    #     clock.tick(30)

    # self.speed = 0
def animateCar():
    rect = screen.get_rect()
    myCar = car("car.png", rect.centerx, rect.centery, 180)

    done = False

    while(1):

```

```

# Event handlers
keys = pygame.key.get_pressed()
clock.tick(fps)
if keys[K_ESCAPE]:
    pygame.quit()
    sys.exit()
elif keys[K_RIGHT]:
    myCar.right()
elif keys[K_LEFT]:
    myCar.left()
elif keys[K_UP]:
    myCar.accelerate()
elif keys[K_DOWN]:
    myCar.decelerate()

myCar.display_car()
pygame.event.pump()

# Game logic
# Clear the screen before drawing
# Display what is drawn:
# pygame.display.flip()

def main():
    abort_criteria = True

    # while abort_criteria:
    #     user_typing = input("Type 1 to escape or 2 to animate car: \n")
    #     if user_typing is "1":
    #         print ("Exiting now")
    #         abort_criteria = False
    #     else:
    #         animateCar()
    animateCar()
main()

```

Пример работы представлен на рисунке 1, где автомобиль можно перемещать клавишами ←↑→↓ по экрану.

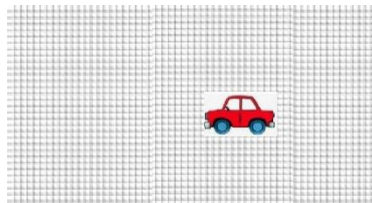


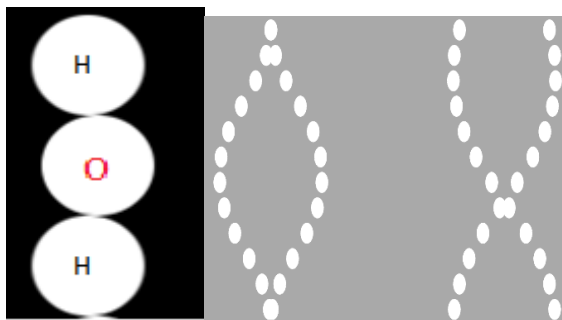
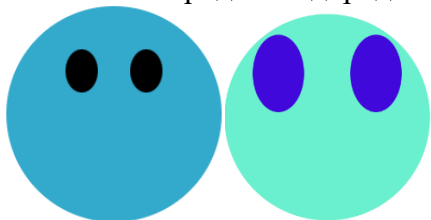
Рисунок 1. Пример работы

На первый взгляд представлено простое программное обеспечение для реализации, но при этом оно помогает понять не только базовые принципы программирования, работу с функциями, правила передачи параметров в функции, но и элементы управления объектом в целом.

Огромным интересом пользуются и анимированные объекты. Здесь также можно предложить множество различных программных пакетов.

Наиболее доступными и простыми являются функции CSS, в которых есть свойство `animation`, которое позволяет анимировать практически любое свойство HTML элементов. Правило анимации указывается в блоке `keyframes`.

Ученикам было представлено задание на создание простейших анимированных объектов, например, создать собственный «смайлик», который моргает глазками, или создать анимированную ДНК – модель, или модель молекулы воды, где происходит «слияние» атомов кислорода и водорода.



```
<html>
<head>
<meta charset="utf-8">
<title>@keyframes</title>
<style>
.creature {
width: 200px; height: 200px;
border-radius: 50%;
background: #3ac;
position: relative;
}
.creature::before, .creature::after {
content: "";
position: absolute;
width: 30px; height: 40px;
border-radius: 50%;
background: #000;
top: 40px;
animation: eye 2s ease-in-out infinite;
}
.creature::before { left: 55px; }
.creature::after { right: 55px; }
@keyframes eye {
90% { transform: none; }
95% { transform: scaleY(0.1); }
}
.btn {
text-decoration: none;
border: 2px solid #764abc;
color: #764abc;
padding: 10px 20px;
border-radius: 25px;
transition: all 1s;
position: relative;

```

```
<body>
<div class="d1"></div>
<div class="d2"></div>
<div class="d3"></div>
<div class="d4"></div>
<div class="d5"></div>
<div class="d6"></div>
</body>
<style>
body{ background-color: black;}
.d1{
width: 50px;
height: 50px;
border-radius: 50%;
background-color: white;
position: relative;
animation: lol 4.5s ease infinite alternate;}
.d2{
width: 50px;
height: 50px;
border-radius: 50%;
background-color: white;
position: relative;
animation: lol 4s ease infinite alternate;}
.d3{
width: 50px;
height: 50px;
border-radius: 50%;
background-color: white;
position: relative;
animation: lol 4.5s ease infinite alternate;}
.d4{
width: 50px;
height: 50px;

```

```

}
.btn::before {
  content: "";
  position: absolute;
  top: 110px; left: 110px;
  width: 100%;
  height: 100%;
  background-color: #764abc;
  transition: all 1s;
}
.btn::before {
  /*...previous code */
  z-index: -1;
  transform: translateX(-100%);
}
.btn:hover::before {
  transform: translateX(0);
}
</style>
</head>
<body>
  <div class="creature"></div>
</body>
</html>
border-radius: 50%;
background-color: white;
position: relative;
animation: lol 4s ease infinite alternate;}
.d5{
  width: 50px;
  height: 50px;
  border-radius: 50%;
  background-color: white;
  position: relative;
  animation: lol 4.5s ease infinite alternate;}
.d6{
  width: 50px;
  height: 50px;
  border-radius: 50%;
  background-color: white;
  position: relative;
  animation: lol 4s ease infinite alternate;}
@keyframes lol{
  from{transform: translate(0,0);}
  to{transform: translate(250px,0);} }
</style>
</html>

```

Представленные примеры являются готовыми алгоритмами, которые можно использовать на практических занятиях по компьютерному моделированию. На основе таких простейших алгоритмических моделей описываются как статичные объекты, так и движущиеся модели с элементами анимации, вращения или перемещения.

Следует отметить, что на практических заданиях по компьютерному моделированию важно сочетать теоретические знания с практическим применением. При формулировке задания четко указывать конкретную цель: что именно обучающиеся должны создать или смоделировать. Важным является то, что цель должна быть достижимой в рамках заданного времени и возможностей. Конкретные примеры и этапы выполнения задания – значимые составляющие в компьютерном моделировании. Задание должно быть мотивирующим, показывать практическое применение компьютерного моделирования.

Осуществление «разбивки» задания на отдельные этапы в начале работы поможет ученикам структурировать процесс работы и «не потеряться» в деталях. На этом этапе возможно обсуждение вопросов и проблем с одноклассниками, предложение обмена идеями. И далее необходимо установить четкие сроки выполнения заданий, что мотивирует учеников и помогает им организовать время. Хорошо организованные занятия по компьютерному моделированию позволяют формировать у учеников такие навыки, как проектирование и планирование, поиск путей решения проблем, коммуникация, алгоритмическое мышление, творческий подход.

Таким образом, использование компьютерных технологий является неотъемлемой частью при формировании алгоритмического мышления студента педагогического вуза. Будущий учитель продумывает последовательность решения задач для достижения поставленных целей, анализирует свои действия и корректирует, разрабатывает алгоритмы. Грамотное проведение практического занятия направлено не только на выполнение заданий, но создает условия для изучения, обсуждения, экспериментирования и развития цифровых компетенций. Компьютерное моделирование – это педагогический инструмент, который

делает процесс обучения более интересным, наглядным и практико-ориентированным, позволяющим активно связывать алгоритмирование с другими изучаемыми дисциплинами.

**Горбунов В.Е., студент 3 курса факультета информационных технологий,
Романенко В.В., канд. физ.-мат. наук, доцент, преподаватель**
Алтайский государственный технический университет им. И.И. Ползунова
г. Барнаул

УЧЕБНЫЕ СИМУЛЯТОРЫ ФИЗИЧЕСКИХ ЗАКОНОВ ДЛЯ ОРГАНИЗАЦИИ ОБУЧЕНИЯ В УНИВЕРСИТЕТЕ НА ПРИМЕРЕ АДИАБАТИЧЕСКОГО РАСШИРЕНИЯ ГАЗА

Аннотация. В настоящей статье представлены симуляторы для проведения лабораторных работ по физике, в соответствии с программой обучения бакалавриата курса «Программной инженерии» Алтайского государственного университета им. И.И. Ползунова. Симуляторы моделируют основные законы динамики поступательного и вращательного движений, явления, связанные с энтропией, и адиабатический процесс. Модели доступны по ссылкам в сети интернет. По ним проходит ознакомление и обучение студентов различных курсов в периоды изучения соответствующих моделям тем. Статья может быть полезна тем, кто заинтересован в возможностях организации и использования более продвинутой системы дистанционного образования, а также создания подобных моделей в других областях наук. Статья избегает вывода физических формул, которые используются в программах, а сам исходный код программ открыт и доступен в рамках лицензии.

Ключевые слова: симулятор, физика, физические законы, методика обучения, информационные технологии, адиабатическое расширение.

**V.E. Gorbunov,
V.V. Romanenko**

TRAINING SIMULATORS OF PHYSICAL LAWS FOR ORGANIZING TRAINING AT THE UNIVERSITY USING THE EXAMPLE OF ADIABATIC EXPANSION OF GAS

Abstract. This article presents simulators for conducting laboratory work in physics, in accordance with the undergraduate curriculum of the "Software Engineering" program at Altai State University named after I.I. Polzunov. The simulators model the fundamental laws of translational and rotational dynamics, phenomena related to entropy and the adiabatic process. The models are available via links on the internet. Students of various courses use them for familiarization and learning during the periods of studying topics corresponding to the models. This article may be useful for those interested in the possibilities of organizing and using a more advanced distance education system, as well as creating similar models in other areas of science. The article avoids stating the physical formulas used in the programs, and the source code of the programs itself is open and available under a license.

Keywords: simulator, physics, physical laws, teaching methodology, information technologies, adiabatic expansion.

Физические симуляторы – программы, которые созданы, чтобы по заданным формулам и алгоритмам моделировать поведение и взаимодействие объектов в системах. Программы симулирования физических законов используют в различных областях, например, в игровых движках (от англ. Game engine), в ракетостроении при подсчете баллистических свойств ракет, также существуют программы, рассчитывающие прочность креплений, конструкций, зоны особых напряжений и т.д. В данной статье авторы