

## ОБУЧЕНИЕ БУДУЩИХ ПЕДАГОГОВ СОЗДАНИЮ ИГРОВЫХ ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ДИНАМИЧЕСКИХ ОБЪЕКТОВ

**Заика И. В. ORCID ID 0000-0001-9154-9565,  
Арапина-Арапова Е. С.**

*Таганрогский институт имени А.П. Чехова (филиал)  
федерального государственного бюджетного образовательного учреждения высшего образования  
«Ростовский государственный экономический университет (РИНХ)»,  
Таганрог, Российская Федерация, e-mail: irinazayka777@gmail.com*

Рост требований к цифровым компетенциям педагогов актуализирует задачу углублённой подготовки будущих учителей в области информационных технологий, включая владение средствами объектно ориентированного программирования. Актуальность исследования обусловлена необходимостью усиления практической подготовки будущих учителей информатики в области объектно ориентированного программирования, особенно в части работы с динамическими объектами. Цель исследования – разработать, теоретически обосновать и экспериментально проверить методический подход к обучению студентов педагогических направлений созданию игровых приложений с динамическими объектами в среде Delphi и разработать практические рекомендации для преподавателя. Материалы исследования: нормативные документы, учебно-методическая литература, авторский комплекс заданий, отчёты студентов, данные педагогического эксперимента. Методы: анализ нормативной и учебно-методической литературы, проектирование комплекса заданий, педагогический эксперимент (констатирующий и контрольный этапы). Разработан комплекс заданий, приведено теоретическое обоснование методического подхода, описаны констатирующий и контрольный эксперименты, а также практические рекомендации по организации поддержки студентов и обратной связи. Апробация на группе студентов второго курса направления «Педагогическое образование» (профили «Математика и Информатика») показала, что большинство участников успешно выполнили задания, средний уровень самостоятельности значительно повысился, а более половины студентов проявили повышенный интерес к объектно ориентированному программированию. Выводы: предложенный подход, включающий комплекс заданий и практические рекомендации, может применяться преподавателями вузов и колледжей при обновлении курсов по программированию, а также в системе дополнительного профессионального образования.

Ключевые слова: студенты, вуз, обучение, программирование, разработка игровых приложений, объектно ориентированное программирование, Delphi, динамические объекты.

## TEACHING FUTURE TEACHERS TO CREATE GAME APPS USING DYNAMIC OBJECTS

**<sup>1</sup>Zaika I. V. ORCID ID 0000-0001-9154-9565,  
<sup>1</sup>Arapina-Arapova E. S.**

*<sup>1</sup>Taganrog Institute named after A.P. Chekhov (branch) of the Federal State Budgetary Educational Institution of Higher Education «Rostov State University of Economics (RINH)», Taganrog, Russian Federation, e-mail: irinazayka777@gmail.com*

Growing demands on teachers' digital competencies make it necessary to provide in-depth training for future teachers in information technology, including object-oriented programming. The relevance of this study stems from the need to strengthen the practical training of future computer science teachers in object-oriented programming, particularly with regard to working with dynamic objects. The objective of this study is to develop, theoretically substantiate, and experimentally test a methodological approach to teaching students majoring in pedagogy to create game applications with dynamic objects in the Delphi environment, and to develop practical recommendations for teachers. Research materials include regulatory documents, educational and methodological literature, the author's set of assignments, student reports, and data from a pedagogical experiment. Methods include analysis of regulatory and educational and methodological literature, design of a set of assignments, and a pedagogical experiment. A set of assignments has been developed, a theoretical justification for the methodological approach (activity-based approach, motivation theory) is provided, the ascertaining and control experiments are described, and practical recommendations for organizing student support and feedback are provided. A pilot study on a group of second-year

students majoring in Teacher Education (majors in Mathematics and Computer Science) revealed that most participants successfully completed the assignments, the average level of independence increased significantly, and more than half of the students demonstrated increased interest in object-oriented programming. Conclusions: The proposed approach, including a set of assignments and practical recommendations, can be used by university and college instructors when updating programming courses, as well as in continuing professional education.

Keywords: students, university, education, programming, development of game applications, object-oriented programming, Delphi, dynamic objects.

## Введение

Стремительный рост требований современного рынка труда к цифровым компетенциям педагогов обуславливает необходимость углублённой подготовки будущих учителей в области информационных технологий, в том числе владения инструментами объектно ориентированного программирования (ООП). Разработка игровых приложений является одним из наиболее эффективных методов повышения мотивации студентов и развития их интереса к информатике. Игровые приложения позволяют студентам осваивать основы объектно ориентированного программирования в увлекательной и доступной форме, создавая визуальные модели реальных процессов и явлений. Хотя основы объектно ориентированного программирования изучаются в рамках школьного курса информатики [1, с. 147-164], практическая подготовка будущих учителей информатики может быть дополнена более глубоким рассмотрением динамических аспектов объектно ориентированного программирования.

Использование Delphi представляет собой актуальное решение для разработки игровых приложений, в которых в качестве игровых элементов применяются динамические объекты [2]. Динамические объекты в Delphi – это экземпляры классов, память под которые распределяется динамически в ходе исполнения программы.

Повышение мотивации и самостоятельности студентов при обучении программированию опирается на деятельностный подход [3, с. 272-284; 4, с. 19-35], согласно которому развитие навыков происходит через активную практическую деятельность. В рамках нашего исследования такой деятельностью выступает создание игрового приложения с динамическими объектами. С позиций деятельностного подхода мотивация формируется в процессе самой деятельности, когда студент видит чёткую цель и имеет возможность поэтапно её достигать [5, с. 45-70]. Согласно теории самодетерминации [6], внутренняя мотивация студентов напрямую зависит от удовлетворения трех базовых психологических потребностей. В разработанном комплексе заданий мы целенаправленно создаем условия для их реализации:

1. Потребность в автономии удовлетворяется через предоставление студентам свободы модификации игровых проектов – выбора дизайна, дополнительных функций, что позволяет им ощущать себя инициаторами и авторами своего обучения.

2. Потребность в компетентности поддерживается принципом постепенного усложнения заданий: от разработки простой игры по образцу до самостоятельной реализации новых механик (например, «Стрельба по мишеням», «Сбор фруктов»). Это создает «лестницу успеха», укрепляя у студента веру в свои силы.

3. Потребность в связанности реализуется через организацию поддерживающей обратной связи от преподавателя.

Анализ литературы позволяет выделить три основных подхода: 1) традиционный (синтаксически ориентированный) – изучение конструкций языка на абстрактных примерах [7, с. 242-246]; 2) проектный – разработка целостного приложения, часто игрового [8]; 3) проблемно ориентированный – решение задач с постепенным наращиванием сложности [9, с. 209-220]. Игровые приложения в Delphi рассматриваются в работе [2; 8].

**Цель исследования** – разработать, теоретически обосновать и экспериментально проверить методический подход к обучению студентов педагогических направлений созданию игровых приложений с динамическими объектами в среде Delphi и разработать практические рекомендации для преподавателя.

#### **Материалы и методы исследования**

Материалами исследования являются: Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 44.03.05 Педагогическое образование (ФГОС ВО) [10] и образовательная программа (ОП) [11] этого же направления, учебно-методическая литература [12, с. 62-73; 13, с. 29-95; 14, с. 7-23], авторский комплекс заданий, отчёты студентов (исходные коды), протоколы наблюдения.

Методы исследования включают: анализ научной и учебно-методической литературы по ООП, изучение нормативной документации, проектирование комплекса заданий, педагогический эксперимент.

Педагогический эксперимент проводился на базе Таганрогского института имени А. П. Чехова (филиал) РГЭУ (РИНХ) в 2024/2025 учебном году. Участники: 16 студентов 2 курса направления 44.03.05 (профили «Математика и Информатика») в рамках дисциплины «Объектно ориентированное программирование». Эксперимент длился 4 недели (8 аудиторных часов и 12 часов самостоятельной работы).

Оценка эффективности проводилась на констатирующем и контрольном этапах по трём критериям: сформированность предварительных навыков программирования (0–3 балла: 0 – не выполнил, 1 – частично с помощью, 2 – выполнил основные функции, 3 – выполнил полностью с дополнительными возможностями); интерес к ООП (анкетирование по 5-балльной шкале, где 5 – очень высокий, 1 – очень низкий); уровень самостоятельности (1 балл

– полное копирование кода, 2 балла – частичная самостоятельная модификация, 3 балла – полная самостоятельная разработка).

*Описание констатирующего этапа.* Перед началом обучения студентам было предложено разработать простое консольное приложение на Delphi без использования динамических объектов (задача: создать массив записей и вывести данные). Результаты: только 3 студента (18,75%) смогли корректно реализовать массивы записей; средний балл по навыку – 1,2; средний интерес – 2,1 (из 5); средняя самостоятельность – 1,2. Отмечены типичные затруднения: непонимание динамического выделения памяти, отсутствие навыков отладки.

*Разработанный комплекс заданий.* Для студентов педагогических направлений предложен и апробирован комплекс из трёх заданий возрастающей сложности.

**Задание 1 (базовое).** Игра «Конфетти» – подробная пошаговая инструкция (см. ниже). Цель – научиться создавать, перемещать и удалять динамические объекты TShape по щелчку мыши, использовать таймер.

**Задание 2 (средней сложности).** «Стрельба по мишеням» – самостоятельная модификация: движение мишеней, подсчёт попаданий, таймер обратного отсчёта.

**Задание 3 (повышенной сложности).** «Сбор фруктов» – полностью самостоятельная разработка с элементами случайного поведения объектов, счётчиком собранных фруктов.

Детальное описание задания 1 «Конфетти» (базовый уровень). Разработать в среде RAD Studio Delphi 10.4 [15] игровое приложение «Конфетти» (рис. 3). Правила: пользователь щёлкает по конфетти левой кнопкой мыши, убирая их с поля за 7 секунд. При успехе выводится «Победа! Молодец!», при неудаче – сообщение о проигрыше.

*Решение.*

1. Создать новый проект: File → New → Windows VCL Application – Delphi. В форме Form1 в Object Inspector свойству Caption присвоить «Конфетти».

2. На форме разместить компоненты: MainMenu1, три Label (Standard), UpDown1 (Win32), Edit1 (Standard), Shape1 (Additional), Timer1 (System). Timer1 – для динамического обновления свойств объектов.

3. Сохранить проект: File → Save Project As. Модуль: UnitGame, проект: ProjectGame.

*Реализация интерфейса.* В таблице 1 показана установка свойств компонентов формы.

Таблица 1

Порядок установки значений свойств компонентов формы «Конфетти»

Выделенный компонент	Окно <i>Object Inspector</i>	Свойство	Установка значений для выделенного компонента
В конструкторе форм выбрать форму «Конфетти» ( <i>Form1</i> )			
Form1	Вкладка Properties	BorderStyle	Выбрать: <i>bsSingle</i>
		ClientHeight	Ввести значение: <i>600</i>

		ClientWidth	Ввести значение: 900
		Color	Выбрать: <i>clSkyBlue</i>
Label1	Вкладка Properties	Align	Выбрать: <i>alTop</i>
		Font	Выбрать размер: 14 Выбрать цвет: <i>зеленый</i>
		Alignment	Выбрать: <i>taCenter</i>
		Align	Выбрать: <i>alTop</i>
Label2	Вкладка Properties	Alignment	Выбрать: <i>taCenter</i>
		Font	Выбрать размер: 48 Выбрать цвет: <i>Красный</i>
		Visible	Выбрать значение: <i>false</i>
		Caption	Ввести текст: <i>Установите время для игры:</i>
Label3	Вкладка Properties	Top	Ввести значение: 165
		Left	Ввести значение: 276
		Font	Выбрать размер: 14
		Top	Ввести значение: 170
Edit1	Вкладка Properties	Left	Ввести значение: 536
		Font	Выбрать размер: 14
		Associate	Выбрать: <i>Edit1</i>
UpDown1	Вкладка Properties	Max	Ввести значение: 20
		Min	Ввести значение: 1
		Visible	Выбрать значение: <i>false</i>
Shape	Вкладка Properties	Enabled	Выбрать значение: <i>false</i>
		Interval	Ввести значение: 500
Timer1	Вкладка Properties	Text	Ввести значение: 7
		ReadOnly	Выбрать значение: <i>true</i>

Примечание: составлено авторами на основе полученных данных в ходе исследования.

Реализация кода модуля *UnitGame*. Нажатием F12 открыть редактор кода. В разделе var объявить глобальные переменные (рис. 1).

```
{1} var
{2} Game1:TForm1;
{3} Obj:TShape;
{4} N, R, I:integer;
{5} T:real;
{6} Log:boolean;
```

Рис. 1. Объявление глобальных переменных. Составлено авторами

В строке 3 – Obj (TShape); строка 4: N – число объектов, R – размер, I – параметр цикла; строка 5: T – время; строка 6: Log – наличие объектов.

Для формы создать процедуру FormCreate (рис. 2).

```
{1} procedure TForm1.FormCreate(Sender: TObject);
{2} begin
{3} Label1.Caption:=' Команда "Начать игру" - запускает игру. '#13#10+
{4} ' Команда "Закрыть игру" - завершает игру. '#13#10+' Правила '+
{5} ' игры: '#13#10+' игрок, щёлкая левой кнопкой мыши по конфетти, должен '+
{6} ' убрать все конфетти '#13#10+' с игрового поля за отведённое время '+
{7} ' (по умолчанию установлено 7 секунд); '#13#10+' если игрок справляется, '+
{8} ' появляется надпись "Победа! Молодец!", иначе '#13#10+
{9} ' выводится сообщение о проигрыше.';
{10} Edit1.Text:='7';
{11} Randomize; N:=15; R:=60; end;
```

Рис. 2. Процедура TForm1.FormCreate. Составлено авторами

Рассмотрим код процедуры TForm1.FormCreate. В строках 3-9 – вывод правил игры; строка 10 – время по умолчанию 7 с; строка 11 – инициализация генератора случайных чисел, N:=15, R:=60.

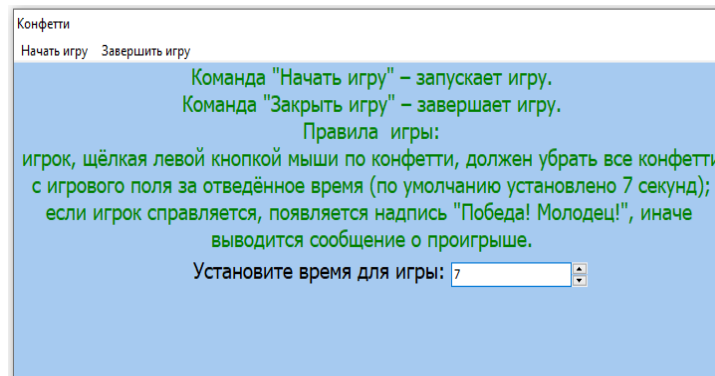


Рис. 3. Внешний вид игрового окна формы «Конфетти». Составлено авторами

Отображение динамических объектов на форме «Конфетти». По команде «Начать игру» срабатывает процедура N1Click (рис. 4).

```
{1} procedure TForm1.N1Click(Sender: TObject);
{2} begin
{3} for I:=1 to N do begin
{4} Obj:=TShape.Create(Form1);
{5} Obj.Name:='Konfetti'+IntToStr(I);
{6} Obj.Width:=R; Obj.Height:=R; Obj.Shape:=stEllipse;
{7} Obj.Top:=Random(Form1.ClientHeight - R);
{8} Obj.Left:=Random(Form1.ClientWidth - R);
{9} Obj.Brush.Color:=Random(10000000);
{10} Obj.OnMouseDown:=ShapelMouseDown;
{11} Obj.Parent:=Form1; end;
{12} Label1.Visible:=false; Label3.Visible:=false;
{13} Edit1.Visible:=false; UpDown1.Visible:=false;
{14} N1.Enabled:=false;
{15} T:= 0; Timer1.Enabled:=true; end;
```

Рис. 4. Процедура TForm1.N1Click. Составлено авторами

Рассмотрим код процедуры TForm1.N1Click. В строках 3-11 цикл создаёт N объектов TShape с уникальными именами, случайными координатами, цветом и обработчиком ShapelMouseDown. Родитель – Form1. В строках 12-15 компоненты Label1, Label3, Edit1, UpDown1 скрываются, команда «Начать игру» блокируется и запускается таймер.

Управление динамическими объектами. Обработчик щелчка мыши ShapelMouseDown (рис. 5).

```
{1} procedure TForm1.ShapelMouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
{2} begin
{3} TShape(Sender).Visible:=false;
{4} Log:=false;
{5} for I:=1 to N do
{6} if TShape(FindComponent('Konfetti'+IntToStr(I))).Visible=true
{7} then Log:=true;
{8} if (not Log) and (T<=StrToInt(Edit1.text)) then begin
```

```

{9} Timer1.Enabled:=false;
{10} Label2.Visible:=true;
{11} Label2.Caption:='Победа! Молодец!';end;end;

```

*Рис. 5. Процедура TForm1.Shape1MouseDown. Составлено авторами*

Рассмотрим код процедуры TForm1.Shape1MouseDown. При щелчке объект исчезает (строка 3). В строках 5-11 цикл проверяет наличие видимых объектов (Log). Если все убраны и время не истекло, таймер останавливается, выводится «Победа! Молодец!».

Для перемещения объектов организуется процедура TForm1.Timer1Timer обработчика события OnTimer (рис. 6).

```

{1}procedure TForm1.Timer1Timer(Sender: TObject);
{2}begin
{3}for I:=1 to N do
{4} with TShape(FindComponent('Konfetti'+IntToStr(I))) do begin
{5} Left:=Left + Random(R div 2)-R div 3;
{6} Top:=Top + Random(R div 2)-R div 3;
{7} if Top < 1 then Top:= 10;
{8} if Left < 1 then Left:=10;
{9} if Top > Form1.ClientHeight - R then Top:=Form1.ClientHeight - R;
{10} if Left > Form1.ClientWidth - R then Left:=Form1.ClientWidth - R; end;
{11} T:=T+(Timer1.Interval)/1000;
{12} Form1.Caption:='Время='+FloatToStr(T)+'с';
{13} if T>StrToInt(Edit1.Text) then begin
{14} Timer1.Enabled:=false;
{15} Label2.Visible:=true; Label2.Caption:='Проигрыш! ';
{16} for I:=1 to N do
{17} TShape(FindComponent('Konfetti'+IntToStr(I))).Enabled:= false; end; end;

```

*Рис. 6. Процедура TForm1.Timer1Timer. Составлено авторами*

Рассмотрим код процедуры TForm1.Timer1Timer. В строках 3-10 цикл случайно изменяет координаты объектов с проверкой границ. Счётчик времени T увеличивается, заголовок формы отображает текущее время. При превышении лимита (строки 13-17) – таймер останавливается, выводится «Проигрыш!», объекты становятся неактивными.

*Завершение игры.* Процедура N2Click для пункта меню «Закреть игру» (рис. 7).

```

{1} procedure TForm1.N2Click(Sender: TObject);
{2} begin
{3} Timer1.Enabled:= false;
{4} for I:=1 to N do (FindComponent('Konfetti'+IntToStr(I))).Free;
{5} N1.Enabled:=true;
{6} Label1.Visible:=true;
{7} Label3.Visible:=true;
{8} UpDown1.Visible:=true;
{9} Edit1.Visible:=true;
{10} Label2.Visible:=false;
{11} Close; end;

```

*Рис. 7. Процедура TForm1.N2Click. Составлено авторами*

Рассмотрим код процедуры TForm1.N2Click. Выполнение кода в строке 3 останавливает работу таймера Timer1. В строке 4 метод Free освобождает место в оперативной

памяти, занятое этим объектом. В строке 5 свойству Enabled объекта N1 присваивается значение true, чтобы пункт меню «Начать игру» стал активным для повторного запуска игры. Аналогично в строках 6-9 свойствам Visible компонент Label1, Label3, UpDown1, Edit1 присваивается значение true. В строке 10 свойству Visible компонента Label2 присваивается значение false.

На рисунке 8 представлено игровое окно, иллюстрирующее игровой процесс.

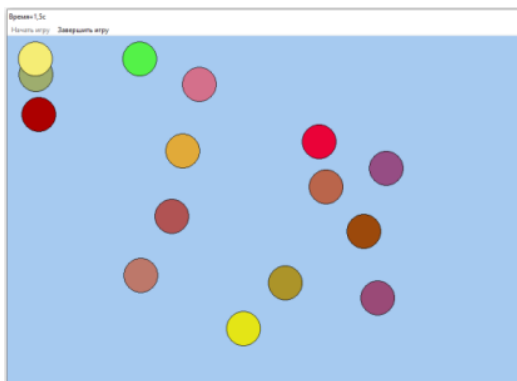


Рис. 8. Иллюстрация игрового процесса. Составлено авторами

#### *Примеры задач для самостоятельного решения*

Задача 1. «Стрельба по мишеням». Разработать игру: пользователь стреляет по движущимся мишеням (стрельба по щелчку мыши) за 10 секунд. Цель – сбить более 10 мишеней. При успехе – «Вы мастер стрельбы!», иначе – «Поражение».

Критерии выполнения: реализовано движение мишеней (по горизонтали/вертикали или произвольной траектории); реализована стрельба по щелчку мыши; реализован таймер обратного отсчёта.

Задача 2. «Сбор фруктов». На поле движутся фрукты (вверх-вниз). Цель – собрать 15 фруктов за 7 секунд (собирать щелчком мыши). При успехе – «Поздравляем! Вы собрали все плоды», при неудаче – «Фрукты упали раньше срока».

Критерии выполнения: фрукты появляются и двигаются случайным образом в вертикальной плоскости; щелчок по фрукту приводит к его исчезновению и увеличению счётчика собранных плодов; реализован таймер, ограничивающий длительность игры; предусмотрена проверка достижения целевого количества фруктов.

#### *Практические рекомендации по организации поддержки студентов и обратной связи.*

Стартовый инструктаж (10 мин.):

1. Объяснение цели и правил игры, демонстрация готового приложения «Конфетти».
2. Пошаговое выполнение с комментированием – преподаватель выполняет код на проекторе, студенты повторяют, задавая вопросы.

3. Поддержка при сложностях: для слабых студентов – карточки-подсказки с фрагментами кода (например, объявление переменных, цикл создания объектов); для средних – чек-лист проверки (Visible настроен? Таймер включён? Обработчик мыши назначен?); для сильных – дополнительные мини-задачи (изменить форму или размер объектов).

4. Обратная связь: после каждого аудиторного занятия – короткий опрос «Что было самым трудным?». Письменные комментарии к каждому сданному проекту в системе электронного обучения в течение 2 дней.

5. Организация парной работы (приветствуется, но с индивидуальной сдачей) для взаимопомощи.

*Результаты контрольного этапа.* После выполнения комплекса заданий проведена итоговая диагностика. Студентам было предложено разработать новое игровое приложение (аналог «Конфетти», но с квадратными объектами и изменённым временем) без готовой инструкции. Сравнение результатов констатирующего и контрольного этапов представлено в таблице 2.

Таблица 2

Сравнение результатов констатирующего и контрольного этапов (средние баллы по группе из 16 человек)

Критерий	Констатирующий этап	Контрольный этап	Прирост	Уровень значимости (p)
Навык работы с динамическими объектами (0–3)	1,2	2,5	+1,3	p < 0,05
Интерес к ООП (1–5)	2,1	4,0	+1,9	p < 0,05
Уровень самостоятельности (1–3)	1,2	2,4	+1,2	p < 0,05

Примечание: составлено авторами на основе полученных данных в ходе исследования.

Положительная динамика зафиксирована по всем критериям. Распределение студентов по уровням самостоятельности изменилось: доля полностью копировавших код сократилась с 68,75% (11 чел.) до 12,5% (2 чел.), тогда как доля выполнявших полную самостоятельную разработку выросла с 6,25% (1 чел.) до 56,25% (9 чел.). Доля студентов с частичной модификацией увеличилась с 25% (4 чел.) до 31,25% (5 чел.). По данным анкетирования, доля студентов с высоким интересом к ООП (4–5 баллов) выросла с 25% до 62,5%.

*Статистическая значимость различий.* Для оценки достоверности изменений использовался парный t-критерий Стьюдента [16, с. 41]. Полученные значения t-критерия для всех трёх показателей превышают критическое значение при уровне значимости 0,05 ( $t(15) > 2,131$ ). Таким образом, различия между констатирующим и контрольным этапами

статистически значимы ( $p < 0,05$ ), что подтверждает эффективность разработанного методического подхода.

### **Результаты исследования и их обсуждение**

Полученные данные подтверждают эффективность предложенного методического подхода. Рост самостоятельности в 2 раза объясняется тем, что пошаговая инструкция игры «Конфетти» формирует когнитивную схему действий, а последующие задания (стрельба, сбор фруктов) требуют её модификации. Можно считать, что предложенный комплекс заданий дополняет существующие методические разработки применительно к педагогическим направлениям подготовки. Вместе с тем ограничением исследования является малая выборка (16 человек); в дальнейшем планируется апробация на больших группах и в дистанционном формате.

### **Заключение**

Проведённое исследование показало, что целенаправленное обучение будущих педагогов созданию игровых приложений с динамическими объектами в Delphi повышает эффективность формирования навыков ООП. Разработанный методический подход включает: теоретическое обоснование (деятельностный подход, теория мотивации); охарактеризованы существующие методы; комплекс из трёх заданий (базовое с полной инструкцией, два – для самостоятельной модификации); детально описанный педагогический эксперимент с количественными данными (уровень самостоятельности вырос с 1,2 до 2,4 балла, доля полностью освоивших – до 75%); практические рекомендации по поддержке студентов и организации обратной связи. Материалы могут быть рекомендованы для педагогических вузов и колледжей. Перспективы дальнейших исследований связаны с адаптацией комплекса для других сред программирования (Python, C#) и дистанционного формата обучения.

### **Список литературы**

1. Семакин И. Г., Хеннер Е. К., Шестакова Л. В. Информатика. 11 класс. Углубленный уровень: учебник: в 2 ч. Ч. 1. М.: БИНОМ. Лаборатория знаний, 2017. 176 с. ISBN: 978-5-9963-3122-2.
2. Заика И. В. Работа с динамическими массивами и текстовыми файлами в визуальной среде разработки программ Delphi // Информатика в школе. 2020. № 4. С. 44-48. DOI: 10.32517/2221-1993-2020-19-4-44-48.
3. Слостенин В. А., Исаев И. Ф., Шиянов Е. Н. Педагогика: учеб. пособие для студ. высш. пед. учеб. заведений / под ред. В. А. Слостенина. М.: Издательский центр «Академия», 2002. 576 с. ISBN: 5-7695-0878-7.

4. Распопова Н. И. Культурно-исторический и деятельностный подход в психологии и образовании: учебное пособие. М.: Русайнс, 2026. 266 с. ISBN: 978-5-466-10524-7.
5. Зимняя И. А. Педагогическая психология: учебник для вузов. 2-е изд., доп., испр. и перераб. М.: Логос, 2009. 384 с. ISBN: 978-5-98704-442-1.
6. Райан Р. М., Деси Э. Л. Теория самодетерминации и поддержка внутренней мотивации, социальное развитие и благополучие // Вестник барнаульского государственного педагогического университета. Издательство: Алтайский государственный педагогический университет. 2003. № 3-1. С. 97-111. EDN: PYQJH.
7. Комлев Н. Ю. Объектно ориентированное программирование. Хорошая книга для хороших людей. М.: Солон-Пресс, 2018. 298 с. ISBN: 978-5-91359-138-8.
8. Емельянов Д. А. Освоение принципов объектно ориентированного программирования в ходе разработки игровых программ // Педагогическое образование в России. 2016. № 7. С. 169-176. DOI: 10.26170/po16-07-24.
9. Кангин В. В., Кангин Е. М. Информатика. Программирование в среде Delphi / под ред. С. Л. Моругина, И. Е. Вострокнутова. Старый Оскол: Тонкие наукоемкие технологии, 2026. 304 с. EDN: FQZUYL, ISBN: 978-5-94178-916-0.
10. Федеральный государственный образовательный стандарт высшего образования – бакалавриат по направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки). URL: [https://fgosvo.ru/uploadfiles/FGOS%20VO%203++/Bak/440305\\_B\\_3\\_16032018.pdf](https://fgosvo.ru/uploadfiles/FGOS%20VO%203++/Bak/440305_B_3_16032018.pdf) (дата обращения 10.03.2026).
11. Образовательная программа по направлению подготовки 44.03.05 Педагогическое образование (профили «Математика и информатика»). URL: <https://www.tgpi.ru/sveden/education/eduprogram> (дата обращения 12.02.2026).
12. Здор Д. В., Горностаева Т. Н. Информатика. Раздел «Программирование». Практикум по программированию в среде Delphi. СПб.: Наукоемкие технологии, 2021. 125 с. EDN: QXCYQX.
13. Васильев Е. П. Среда визуального программирования Delphi: теория и практика. Учебное пособие для бакалавров высших учебных заведений, обучающихся по направлению подготовки 02.03.01 "Математика и компьютерные науки" (квалификация бакалавр). Рязань: Book jet, 2019. 204 с. EDN: OESSMP.
14. Ключникова Т. А., Меркурьева Е. Д., Шайхлисламова С. В. Изучение компонентов ввода-вывода пакета Delphi в рамках курса «Алгоритмизация и программирование». М.: РГУ нефти и газа имени И. М. Губкина, 2022. 82 с. EDN: NZBSCL.

15. Embarcadero DocWiki. Официальная документация Delphi. URL: <https://docwiki.embarcadero.com>. (дата обращения: 15.04.2026).

16. Афанасьев В. В., Сивов М. А. Математическая статистика в педагогике: учебное пособие. Ярославль: Изд-во ЯГПУ, 2010. 75 с. ISBN: 978-5-87555-366-0.

**Конфликт интересов:** Авторы заявляют об отсутствии конфликта интересов.

**Conflict of interest:** The authors declare that there is no conflict of interest.

**Финансирование:** Авторы заявляют об отсутствии внешнего финансирования.

**Financing:** The research was performed without external funding.