МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ Федеральное государственное бюджетное образовательное учреждение высшего образования «ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» Физический факультет

#### РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

#### Компьютерное моделирование в физике

Общей и теоретической физики

Физического факультета

#### Образовательная программа 03.03.02 ФИЗИКА

Профиль подготовки – «Фундаментальная физика»

Уровень высшего образования – Бакалавриат

Форма обучения – очная

Статус дисциплины: вариативная

Махачкала, 2020 год

Рабочая программа дисциплины «Компьютерное моделирование в физике» составлена в 2020 году в соответствии с требованиями ФГОС ВО по направлению подготовки 03.03.02 – Физика (уровень: бакалавриат) от «\_7\_» августа 2014г. № 937.

Разработчик(и): кафедра Общей и теоретической физики Магомедов М.А. к.ф.-м.н., доцент

Рабочая программа дисциплины одобрена: на заседании кафедры общей и теоретической физики от «21» января 2020 г., протокол № 5

Зав. кафедрой

Attachel-

Муртазаев А.К.

На заседании Методической комиссии Физического факультета от «28 »февраля 2020 г., протокол №6

Председатель

<u>Муриен</u> Мурлиева Ж.Х.

Рабочая программа дисциплины согласована с учебнометодическим управлением «26» марта 2020г. .

Начальник УМУ

the

Гасангаджиева А.Г

#### Аннотация рабочей программы дисциплины

Дисциплина «Компьютерное моделирование в физике» входит в *вариативную* часть по выбору образовательной программы *бакалавриата* по направлению 03.03.02 «Физика»

Дисциплина реализуется на физическом факультете кафедрой общей и теоретической физики.

Содержание дисциплины охватывает круг вопросов, связанных с основами компьютерного моделирования различных физических явлений.

Дисциплина нацелена на формирование следующих компетенций выпускника:

Общепрофессиональных: - ОПК -5 профессиональных – **ПК-5**.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лабораторные занятия, самостоятельная работа.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости в форме контрольная работа и промежуточный контроль в форме зачета.

Объем дисциплины 2 зачетных единиц, в том числе в академических часах по видам учебных занятий – 72 часа.

Семес			Форма					
тр			промежуточной					
	Ко	нтактная	аттестации					
	Bce			в том	(зачет,			
	го	Лекц	Лаборатор	Практич	КСР	консульт	числе	дифференцирова
		ИИ	ные	еские		ации	экзам	нный зачет,
			занятия	занятия			ен	экзамен
6	72	16					56	зачет

#### 3. Цели освоения дисциплины

Целью освоения дисциплины «Компьютерное моделирование в физике» является знакомство студентов с основными компьютерного моделирования различных физических явлений, различными численными методами и реализующими их алгоритмами, а также подготовка студентов к решению практических задач с помощью компьютерного моделирования.

Ускорение научно-технического процесса, проникновение ЭВМ во все сферы деятельности человека, повышение роли ЭВМ в фундаментальных и прикладных исследованиях связи с необходимостью широкого использования математических моделей и компьютерного моделирования.

Таким образом, дисциплина «Компьютерное моделирование в физике» имеет целью:

- Ознакомить студентов с методами компьютерного моделирования в физике;
- научить студентов разработке математических моделей физических объектов;
- дать навыки постановки численного эксперимента;
- ознакомить с методами обработки и интерпретации результатов компьютерного моделирования.

В курсе излагаются основы компьютерного моделирования, методы и алгоритмы вычислительной физики и способы их математического моделирования.

Курс включает лекционные занятия.

#### 2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Компьютерное моделирование в физике» входит в базовый компонент цикла естественнонаучных и математических (ЕН и М) дисциплин и является обязательной для изучения.

Для изучения дисциплины «Компьютерное моделирование в первоначальные физике» студент должен знать: знания из курсов линейной математического анализа. алгебры. обыкновенных дифференциальных уравнений, уравнений математической физики. Знания и умения, практические навыки, приобретенные студентами в результате дисциплины, использоваться изучении изучения будут при курсов математического моделирования, вычислительного практикума, при выполнении курсовых и дипломных работ, связанных с математическим моделированием и обработкой наборов данных, решением конкретных задач из механики, физики и т.п.

дисциплины (	перечень планируемых	результатов обучения).
	Формулировка	Планируемые результаты обучения
Компетенции	компетенции из ФГОС	(показатели достижения заданного
	BO	уровня освоения компетенций)
ОПК-5	способностью использовать основные методы, способы и средства получения, хранения, переработки информации и навыки работы с компьютером как со средством управления информацией;	Знать: Теоретические основы и современные методы и способы получения, хранения и и переработки информации. Уметь: С помощью персонального компьютера хранить и передавать информацию; Считывать и записывать информацию; Считывать и записывать информации; Использовать различные методы передачи и обработки информации. Владеть: Общими методами хранения и обработки научной информации; современным персональным компьютером; современными средствами получения, хранения и передачи информации; информационными и интернет- технологиями.
ПК-5	способностью пользоваться современными методами обработки, анализа и синтеза физической информации в избранной области физических исследований	Знать: Теоретические основы и специальный математический аппарат компьютерного моделирования физических систем; Способы реализации и программирования различных вычислительных методов. Уметь: Построить математические модели физических систем; Ставить численный эксперимент; использовать различные математические модели при компьютерном моделировании различных физических систем. Владеть: Общими методами построения математических моделей для компьютерного моделирования различных физических явлений. Методами анализа результатов компьютерного моделирования с использованием специализированного программного обеспечения.

## 3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения).

#### 4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет \_2\_\_\_ зачетных единиц, \_\_72\_\_ академических часов.

#### 4.2. Структура дисциплины.

№ п/п	Разделы и темы дисциплины		естра	Вид са ра	ы учеб вкл амосто боту ст грудоем ча	ной ра ючая ятельн гуденто икость сах)	боты, ую ов и (в	ая работа	Формы текущего контроля успеваемости <i>(по неделям семестра)</i> Форма промежуточной
		Семест Неделя сем		Лекции	Практические занятия	Лабораторные занятия	Контроль самост. раб.	Самостоятельн	аттестации (по семестрам)
	Модуль 1.								
1.	Введение. Компьютерное моделирование в среде Delphi 7. Знакомство со средой программирования. Создание простейших программ.	6		2				7	Проверка выполнения лабораторной работы
2.	Построение графиков функций в среде Delphi. Эпициклоида, Эпитрохоида, Гипотрохоида.	6		2				7	Проверка выполнения лабораторной работы
3.	Компьютерное моделирование движения тела в гравитационном поле.	6		2				7	Проверка выполнения лабораторной работы
4.	Компьютерное моделирование движения планет солнечной системы.	6		2				7	Проверка выполнения лабораторной работы
	ВСЕГО ЗА МОДУЛЬ 1.			8				28	
	Модуль 2.								

5.	Компьютерное моделирование идеального газа в сосуде.	6	3		10	Проверка выполнения лабораторной работы
6.	Компьютерное моделирование эффекта перколяции.	6	3		10	Проверка выполнения лабораторной работы
7.	Компьютерное моделирование фрактальных систем.	6	2		8	Проверка выполнения лабораторной работы
	ВСЕГО ЗА МОДУЛЬ 2.		8		28	
	ИТОГО:		16		56	

#### 4.3. Содержание дисциплины, структурированное по темам (разделам).

#### Модуль 1.

Тема 1. Введение. Компьютерное моделирование в среде Delphi 7. Знакомство со средой программирования. Создание простейших программ.

#### Структура среды программирования

Внешний вид среды программирования Delphi отличается от многих других из тех, что можно увидеть в Windows. К примеру, Borland Pascal for Windows 7.0, Borland C++ 4.0, Word for Windows, Program Manager - это все MDI приложения и выглядят по-другому, чем Delphi. MDI (Multiple Document Interface) - определяет особый способ управления нескольких дочерних окон внутри одного большого окна.

Среда Delphi же следует другой спецификации, называемой Single Document Interface (SDI), и состоит из нескольких отдельно расположенных окон. Это было сделано из-за того, что SDI близок к той модели приложений, что используется в Windows 95.

Если Вы используете SDI приложение типа Delphi, то уже знаете, что перед началом работы лучше минимизировать другие приложения, чтобы их окна не загромождали рабочее пространство. Если нужно переключиться на другое приложение, то просто щелкните мышкой на системную кнопку минимизации Delphi. Вместе с главным окном свернутся все остальные окна среды программирования, освободив место для работы других программ.

#### Главные составные части среды программирования

Ниже перечислены основные составные части Delphi:

- 1. Дизайнер Форм (Form Designer)
- 2. Окно Редактора Исходного Текста (Editor Window)
- 3. Палитра Компонент (Component Palette)
- 4. Инспектор Объектов (Object Inspector)
- 5. Справочник (On-line help)

Есть, конечно, и другие важные составляющие Delphi, вроде линейки инструментов, системного меню и многие другие, нужные Вам для точной настройки программы и среды программирования.

Программисты на Delphi проводят большинство времени переключаясь между Дизайнером Форм и Окном Редактора Исходного Текста (которое для краткости называют Редактор). Прежде чем Вы начнете, убедитесь, что можете распознать эти два важных элемента. Дизайнер Форм показан на рис.1, окно Редактора - на рис.2.

For	m1 _	. 🗆 ×
	• • • • • • • • • • • • • • • • • • • •	

Рис.1: Дизайнер Форм - то место, где Вы создаете визуальный интерфейс программы.

🗎 UNIT1.PAS		_ 🗆 ×
<b>unit</b> Unit1	L;	
interface		_
uses		
SysUtils	3, WinTypes, WinProcs, Messages, Classes	,
Graphics	3, Controls, Forms, Dialogs;	
type		
TForm1 =	• <b>class</b> (TForm)	
private		
{ Priv	vate declarations }	
public	·	
( Publ	lic declarations }	
end;		
var		
Form1: T	(Form1;	
implementa	ition	-
7: 22 Modifi	ied Insert 🖣	•
\Unit1 /		

Рис.2: В окне Редактора Вы создаете логику управления программой.

Дизайнер Форм в Delphi столь интуитивно понятен и прост в использовании, что создание визуального интерфейса превращается в детскую игру. Дизайнер Форм первоначально состоит из одного пустого окна, которое Вы заполняете всевозможными объектами, выбранными на Палитре Компонент.

Несмотря на всю важность Дизайнера Форм, местом, где программисты проводят основное время является Редактор. Логика является движущей силой программы и Редактор - то место, где Вы ее "кодируете".

Палитра Компонент (см. рис.3) позволяет Вам выбрать нужные объекты для размещения их на Дизайнере Форм. Для использования Палитры Компонент просто первый раз щелкните мышкой на один из объектов и потом второй раз - на Дизайнере Форм. Выбранный Вами объект появится на проектируемом окне и им можно манипулировать с помощью мыши.

Палитра Компонент использует постраничную группировку объектов. Внизу Палитры находится набор закладок - Standard, Additional, Dialogs и т.д. Если Вы щелкнете мышью на одну из закладок, то Вы можете перейти на следующую страницу Палитры Компонент. Принцип разбиения на страницы широко используется в среде программирования Delphi и его легко можно использовать в своей программе. (На странице Additional есть компоненты для организации страниц с закладками сверху и снизу).



Рис.3: Палитра Компонент - место, где Вы выбираете объекты, которые будут помещены на вашу форму.

Предположим, Вы помещаете компонент TEdit на форму; Вы можете двигать его с места на место. Вы также можете использовать границу, прорисованную вокруг объекта для изменения его размеров. Большинством других компонент можно манипулировать тем же образом. Однако, невидимые во время выполнения программы компоненты (типа TMenu или TDataBase) не меняют своей формы.

Слева от Дизайнера Форм Вы можете видеть Инспектор Объектов (рис.4). Заметьте, что информация в Инспекторе Объектов меняется в зависимости от объекта, выбранного на форме. Важно понять, что каждый компонент является настоящим объектом и Вы можете менять его вид и поведение с помощью Инспектора Объектов.

Инспектор Объектов состоит из двух страниц, каждую из которых можно использовать для определения поведения данного компонента. Первая страница - это список свойств, вторая - список событий. Если нужно изменить что-нибудь, связанное с определенным компонентом, то Вы обычно делаете это в Инспекторе Объектов. К примеру, Вы можете изменить имя и размер компонента TLabel изменяя свойства Caption, Left, Top, Height, и Width.

Вы можете использовать закладки внизу Инспектора Объектов для переключения между страницами свойств и событий.

🧏 Object Insp	ector 💶 🗵 🗙				
Form1: TForm1	•				
ActiveControl					
AutoScroll	True				
+BorderIcons	[biSystemMenu,biMinim				
BorderStyle	bsSizeable				
Caption	Form1				
ClientHeight	273				
ClientWidth	427				
Color	clBtnFace				
CtI3D	True				
Cursor	crDefault				
Enabled	True				
+Font	(TFont)				
FormStyle	fsNormal				
Height	300				
HelpContext	0				
Hint					
+HorzScrollBar	(TControlScrollBar)				
lcon	(None)				
KeyPreview	False				
Left	463 💌				
Properties (Events /					

Рис.4: Инспектор Объектов позволяет определять свойства и поведение объектов, помещенных на форму.

Страница событий связана с Редактором; если Вы дважды щелкнете мышкой на правую сторону какого-нибудь пункта, то соответствующий данному событию код автоматически запишется в Редактор, сам Редактор немедленно получит фокус, и Вы сразу же имеете возможность добавить код обработчика данного события. Данный аспект среды программирования Delphi будет еще обсуждаться позднее.

#### Стандартные компоненты

Для дальнейшего знакомства со средой программирования Delphi потребуется рассказать о составе первой страницы Палитры Компонент.

На первой странице Палитры Компонент размещены 14 объектов (рис.8) определенно важных для использования. Мало кто обойдется длительное время без кнопок, списков, окон ввода и т.д. Все эти объекты такая же часть Windows, как мышь или окно.

Набор и порядок компонент на каждой странице являются конфигурируемыми. Так, Вы можете добавить к имеющимся компонентам новые, изменить их количество и порядок.

R	( 5	<mark>الم</mark>	Α	<u>ab]]</u>		OK	×	۲	<b>.</b>	Ē					$\rangle$
\Stan	dard (Ad	dition	ial (D	ata Ac	cess	∬Data	a Cont	rols (	Dialog	38 <b>(</b> S J	/stem/	(VBX	/(Sam	ples/	

Рис.8: Компоненты, расположенные на первой странице Палитры.

Стандартные компоненты Delphi перечислены ниже с некоторыми комментариями по их применению. При изучении данных компонент было бы полезно иметь под рукой компьютер с тем, чтобы посмотреть, как они работают и как ими манипулировать.

- **ТМаіпМепи** позволяет Вам поместить главное меню в программу. При помещении TMainMenu на форму это выглядит, как просто иконка. Иконки данного типа называют "невидимыми компонентом", поскольку они невидимы во время выполнения программы. Создание меню включает три шага: (1) помещение TMainMenu на форму, (2) вызов Дизайнера Меню через свойство Items в Инспекторе Объектов, (3) определение пунктов меню в Дизайнере Меню.
- **ТРорирМени** позволяет создавать всплывающие меню. Этот тип меню появляется по щелчку правой кнопки мыши.
- **TLabel** служит для отображения текста на экране. Вы можете изменить шрифт и цвет метки, если дважды щелкнете на свойство Font в Инспекторе Объектов. Вы увидите, что это легко сделать и во время выполнения программы, написав всего одну строчку кода.
- **TEdit** стандартный управляющий элемент Windows для ввода. Он может быть использован для отображения короткого фрагмента текста и позволяет пользователю вводить текст во время выполнения программы.
- **ТМето** иная форма TEdit. Подразумевает работу с большими текстами. ТМето может переносить слова, сохранять в Clipboard фрагменты текста и восстанавливать их, и другие основные функции редактора. ТМето имеет ограничения на объем текста в 32Кб, это
- составляет 10-20 страниц. (Есть VBX и "родные" компоненты Delphi, где этот предел снят).
- **TButton** позволяет выполнить какие-либо действия при нажатии кнопки во время выполнения программы. В Delphi все делается очень просто. Поместив TButton на форму, Вы по двойному щелчку можете создать заготовку обработчика события нажатия кнопки. Далее нужно заполнить заготовку кодом (подчеркнуто то, что нужно написать вручную):
- **TCheckBox** отображает строку текста с маленьким окошком рядом. В окошке можно поставить отметку, которая означает, что что-то выбрано. Например, если посмотреть окно диалога настроек компилятора (пункт меню Options | Project, страница Compiler), то можно увидеть, что оно состоит преимущественно из CheckBox'ов.
- **TRadioButton** позволяет выбрать только одну опцию из нескольких. Если Вы опять откроете диалог Options | Project и выберете страницу Linker

Options, то Вы можете видеть, что секции Map file и Link buffer file состоят из наборов RadioButton.

- **TListBox** нужен для показа прокручиваемого списка. Классический пример ListBox'а в среде Windows выбор файла из списка в пункте меню File | Ореп многих приложений. Названия файлов или директорий и находятся в ListBox'e.
- **ТСотвоВох** во многом напоминает ListBox, за исключением того, что позволяет водить информацию в маленьком поле ввода сверху ListBox. Есть несколько типов ComboBox, но наиболее популярен выпадающий вниз (drop-down combo box), который можно видеть внизу окна диалога выбора файла.
- **TScrollbar** полоса прокрутки, появляется автоматически в объектах редактирования, ListBox'ах при необходимости прокрутки текста для просмотра.
- **TGroupBox** используется для визуальных целей и для указания Windows, каков порядок перемещения по компонентам на форме (при нажатии клавиши TAB).
- **TPanel** управляющий элемент, похожий на TGroupBox, используется в декоративных целях. Чтобы использовать TPanel, просто поместите его на форму и затем положите другие компоненты на него. Теперь при перемещении TPanel будут передвигаться и эти компоненты. TPanel используется также для создания линейки инструментов и окна статуса.
- **TScrollBox** представляет место на форме, которое можно скроллировать в вертикальном и горизонтальном направлениях. Пока Вы в явном виде не отключите эту возможность, форма сама по себе действует так же. Однако, могут быть случаи, когда понадобится прокручивать только часть формы. В таких случаях используется TScrollBox.

Это полный список объектов на первой странице Палитры Компонент. Если Вам нужна дополнительная информация, то выберите на Палитре объект и нажмите клавишу F1 - появится Справочник с полным описанием данного объекта.

#### Сохранение программы

Вы приложили некоторые усилия по созданию программы и можете захотеть ее сохранить. Это позволит загрузить программу позже и снова с ней поработать.

Первый шаг - создать поддиректорию для программы. Лучше всего создать директорию, где будут храниться все Ваши программы и в ней - создать поддиректорию для данной конкретной программы. Например, Вы можете создать директорию MYCODE и внутри нее - вторую директорию TIPS1, которая содержала бы программу, над которой Вы только что работали.

После создания поддиректории для хранения Вашей программы нужно выбрать пункт меню File | Save Project. Сохранить нужно будет два файла. Первый - модуль (unit), над которым Вы работали, второй - главный файл

проекта, который "владеет" Вашей программой. Сохраните модуль под именем MAIN.PAS и проект под именем TIPS1.DPR. (Любой файл с расширением PAS и словом "unit" в начале является модулем.)

## Тема 2. Построение графиков функций в среде Delphi. Эпициклоида, Эпитрохоида, Гипотрохоида.



Создайте приложение **Delphi** и разместите на форме следующие элементы:

- **1. Button1** (с надписью Старт!!!).
- 2. Checkbox1 с надписью STOP. Checkbox2 с надписью PAUSE.
- 3. Label1, Label2, Label3, Label4 с надписями R1=, R2=, dt=, h=.
- 4. Edit1, Edit2, Edit3, Edit4. Присвоить значения 3, 1, 0.001, 0.5.
- **5. СотвоВох1.** В свойство **СотвоВох1.Items** добавить
  - 1. Эпициклоида
  - 2. Эпитрохоида
  - 3. Гипотрохоида

#### 6. Chart1.

На элемент Chart1 добавить 6 кривых

кривые 1, 3, 4 и 6 – **line**, 2 и 5 – **Points**.

Задать следующие свойства кривой:

В Object Tree View выделить Chart1.Series1 и в инспекторе объектов Object Inspector установить для Xvalues.Order значение IoNone Повторить то же самое для Series3, Series4 и Series6.

Chart1.Title	Убираем галочку	Visible
Chart1.Legend	Убираем галочку	Visible

Дважды кликаем по пустому месту на форме и набираем первый блок кода программы:

Дважды кликаем по кнопке СТАРТ! и набираем 2 блок кода программы: Код программы

```
//======
unit Unit1:
               !!! Пропуск интерфейсной части
{$R *.dfm}
_____
procedure TForm1.FormCreate(Sender: TObject);
begin
   Decimalseparator := '.';
   Combobox1.ItemIndex := 0;
end:
//== 2 =======
                        _____
procedure TForm1.Button1Click(Sender: TObject);
var
  x,y,x2,y2,t,t2,tmax,dt,r1,r2,k,h:Real;
  iv:Integer;
begin
   DecimalSeparator := '.';
   CheckBox1.Checked := False;
   CheckBox2.Checked := False;
   r1 := StrToFloat(Edit1.Text);
   r2 := StrToFloat(Edit2.Text);
   dt := StrToFloat(Edit3.Text);
   h := StrToFloat(Edit4.Text);
  if ComboBox1.ItemIndex = 0 then h := r2; // Эпициклоида
   Chart1.Series[0].Clear;
                            Chart1.Series[1].Clear;
                                                     Chart1.Series[2].Clear;
   iv := 0;
// Рисуем внутренний круг
   t := 0;
repeat
   x := r1*Cos(t); y := r1*Sin(t);
   Chart1.Series[2].AddXY(x,y,' ',clBlue);
   t := t + 0.01;
until t > 2*pi;
       Application.ProcessMessages;
   t := 0;
repeat
Case Combobox1.ItemIndex of
0,1:
  begin
       x := r1*(r2/r1 + 1)*Cos(t*r2/r1) - h*Cos((r2/r1 + 1)*t);
        y := r1*(r2/r1 + 1)*Sin(t*r2/r1) - h*Sin((r2/r1 + 1)*t);
  end;
2:
  begin
         x := r1*(r2/r1 + 1)*Cos(t*r2/r1) - h*Cos((r2/r1 + 1)*t);
         y := r1^{*}(r2/r1 + 1)^{*}Sin(t^{*}r2/r1) - h^{*}Sin((r2/r1 + 1)^{*}t);
         x := (r1-r2)*Cos(t) + h*Cos((r1-r2)*t/r2);
```

```
y := (r1-r2)*Sin(t) - h*Sin((r1-r2)*t/r2);
  end;
end;
    Chart1.Series[0].AddXY(x,y);
    t := t + dt;
    Inc(iv);
    if iv > 100 then
    begin
        iv := 0;
        dt := StrToFloat(Edit3.Text);
    Chart1.Series[1].Clear;
                                Chart1.Series[1].AddXY(x,y);
// Рисуем внешний круг
    t2 := 0;
    Chart1.Series[3].Clear;
                                Chart1.Series[4].Clear;
                                                           Chart1.Series[5].Clear;
repeat
Case Combobox1.ItemIndex of
0,1:
  begin
        x2 := r1*(r2/r1 + 1)*Cos(t*r2/r1) + r2*Cos(t2);
        y_2 := r_1 (r_2/r_1 + 1) Sin(t r_2/r_1) + r_2 Sin(t_2);
  end;
2:
  begin
        x2 := (r1-r2)*Cos(t) - r2*Cos(t2);
        y_2 := (r_1 - r_2) * Sin(t) - r_2 * Sin(t_2);
  end;
end;
    Chart1.Series[3].AddXY(x2,y2,",clgreen);
    t2 := t2 + 0.01;
until t^2 > 2^* pi;
Case Combobox1.ItemIndex of
0.1:
  begin
        x2 := r1*(r2/r1 + 1)*Cos(t*r2/r1);
        y_2 := r_1 (r_2/r_1 + 1) Sin(t r_2/r_1);
  end;
2:
  begin
        x2 := (r1-r2)*Cos(t);
        y_2 := (r_1 - r_2) * Sin(t);
  end;
end;
    Chart1.Series[4].AddXY(x2,y2,",clBlack);
    Chart1.Series[5].AddXY(x2,y2,",clBlack);
    Chart1.Series[5].AddXY(x,y,",clBlack);
        Application.ProcessMessages;
    end:
// Установка Паузы при рисовании
if CheckBox2.Checked then
    repeat Application.ProcessMessages;
    until
           not(CheckBox2.Checked);
until CheckBox1.Checked;
```

end; end. Результаты выполнения программы:



## 

Укороченная Эпитрохоида



Удлиненная Эпитрохоида





Улитка Паскаля

Роза

## **Тема 3.** Компьютерное моделирование движения тела в гравитационном поле.

Создайте новый проект Delphi. Разместите на нем следующие объекты: Кнопку Button1 Поля для ввода стартовых значений Edit1, Edit2, Edit3, Edit4, Edit5 Метки для обозначения полей Label1, ..., Label5 4 Графика Chart1, Chart2, Chart3, Chart4 (элемент Chart находится в вкладке Additional) Для Chart1 добавить 2 кривые, одна – точки, вторая - линия. Поле для вывода Memo1



Рис 7.1. Рабочее окно программы



Рис 7.2. результат выполнения программы.

Дважды нажать на кнопку **Button1** и в появившемся окне дописать процедуру:

```
procedure TForm1.Button1Click(Sender: TObject);
var
   i,tv:Integer;
   h,x,y,vx,vy,v,ax,ay,a,alfa,t,dt,ymax,xmax,vmax:Real;
begin
   DecimalSeparator := '.';
   h := StrToFloat(Edit1.Text);
   v := StrToFloat(Edit2.Text);
 alfa := StrToFloat(Edit3.Text)*pi/180;
   dt := StrToFloat(Edit4.Text);
   tv := Trunc(StrToFloat(Edit5.Text)*10);
   vx := v*Cos(alfa); vy := v*Sin(alfa);
   ax := 0;
           ay := -9.81;
   t := 0;
//-----
 Chart1.Series[0].Clear; Chart1.Series[1].Clear; Chart2.Series[0].Clear;
 Chart3.Series[0].Clear; Chart4.Series[0].Clear;
 Memo1.Clear;
//-----
         x := 0;
                     \mathbf{y} := \mathbf{h};
     xmax := x;
                    ymax := y; ymax := v;
 repeat
       x := x + vx^*dt;
        y := y + vy^*dt;
      vx := vx + ax^*dt;
      vy := vy + ay^*dt;
      v := sqrt(vx*vx + vy*vy);
      t := t + dt;
//-----
 Chart1.Series[0].Clear;
 Chart1.Series[0].AddXY(x,y);
                                      sleep(tv);
 Chart1.Series[1].AddXY(x,y);
 Chart2.Series[0].AddXY(t,vx);
 Chart3.Series[0].AddXY(t,vy);
 Chart4.Series[0].AddXY(t,v);
//-----
      if x \max < x then x \max := x;
      if ymax < y then ymax := y;
      if vmax < v then vmax := v;
//-----
  Application.ProcessMessages;
 Until y < 0;
//-----
     Memo1.Lines.Add('xmax = ' + floattostrf(xmax,fffixed,7,5));
     Memo1.Lines.Add('ymax = ' + floattostrf(ymax,fffixed,7,5));
     Memo1.Lines.Add('vmax = ' + floattostrf(vmax,fffixed,7,5));
     Memo1.Lines.Add('tmax = ' + floattostrf(t,fffixed,7,5));
end;
end.
```

Дополнить программу по мере необходимости.

### Тема 4. Компьютерное моделирование движения планет солнечной системы.



Создайте новое приложение **Delphi** и разместите на форме следующие элементы

- **1. Button1** (с надписью Старт!!!)
- 2. Chart1
- 3. Checkbox1 с надписью Pause, Checkbox2 с надписью STOP
- 4. Edit1, Edit2, Edit3, Edit4, Edit5. Присвоить значения 0.85, 1, 0.0001, 0.2, 3 соответственно.
- **5.** Label1, Label2, Label3, Label4, Label5 с надписями v=, r=, dt=, v2=, r2 соответственно

На элемент **Chart1** добавить 5 кривых, 1, 2 и 4 кривая – Points, 3 и 5 кривая – line Задать следующие свойства кривых:

В Object Tree View выделить Chart1.Series3 и в инспекторе объектов Object Inspector установить для Xvalues.Order значение ІоNопе В Object Tree View выделить Chart1.Series5 и в инспекторе объектов Object Inspector установить для Xvalues.Order значение ІоNопе Для графика Chart1 убираем галочки в пунктах: Chart1.Title Убираем галочку Visible Chart1.Legend Убираем галочку Visible кнопка "Grid Border..." Chart1.Axis.Left.Ticks. Убираем галочку Visible Chart1.Axis.Bottom.Ticks. кнопка "Grid Border..." Убираем галочку Visible

#### Код программы

```
unit Unit1;
              !!! Пропуск интерфейсной части
{$R *.dfm}
//========= При запуске приложения
                                                   =======
procedure TForm1.FormCreate(Sender: TObject);
begin
    DecimalSeparator := '.';
end;
Моделирование
                                              ============
procedure TForm1.Button1Click(Sender: TObject);
var
   x,y,vx,vy,ax,ay,t,dt,r:real;
   xmin1,xmax1,ymin1,ymax1,t1,a1,b1,e1:real;
   x2,y2,vx2,vy2,ax2,ay2,r2,r21:real;
   i,iv:integer;
begin
   checkbox2.checked := False;
   Chart1.Series[0].Clear;
   Chart1.Series[1].Clear;
   Chart1.Series[2].Clear;
   Chart1.Series[3].Clear;
   Chart1.Series[4].Clear;
        iv := 0;
    x := StrToFloat(Edit2.Text);
    y := 0;
   vx := 0.0;
   vy := StrToFloat(Edit1.Text);
   vy2 := StrToFloat(Edit4.Text);
   x2 := StrToFloat(Edit5.Text);
   y2 := 0;
   vx2 := 0.0;
   vy2 := StrToFloat(Edit4.Text);
   r21 := x2/x;
    t := 0;
   dt := StrToFloat(Edit3.Text);
   Chart1.Series[0].AddXY(0,0,",clred);
   Chart1.Series[1].AddXY(x,y);
        Chart1.Series[2].AddXY(x,y);
        Chart1.Series[2].Clear;
        Chart1.Series[2].AddXY(x,y);
   xmin1 := 0; xmax1 := 0;
   ymin1 := 0;
                ymax1 := 0;
```

```
repeat
                dt := StrToFloat(Edit3.Text);
                     ax := -x/sqrt((x^*x + y^*y)^*(x^*x + y^*y)^*(x^*x + y^*y));
                     ay := -y/sqrt((x^*x + y^*y)^*(x^*x + y^*y)^*(x^*x + y^*y));
                     vx := vx + ax^*dt;
                     vy := vy + ay^{*}dt;
                       x := x + vx^*dt;
                       y := y + vy^*dt;
                     ax2 := -x2/sqrt((x2*x2 + y2*y2)*(x2*x2 + y2*y2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*
y2*y2))/r21;
                     ay2 := -y2/sqrt((x2*x2 + y2*y2)*(x2*x2 + y2*y2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*x2)*(x2*
y2*y2))/r21;
                     vx2 := vx2 + ax2*dt;
                     vy2 := vy2 + ay2*dt;
                       x2 := x2 + vx2*dt;
                       y2 := y2 + vy2*dt;
                     if x < xmin1 then xmin1 := x;
                     if x > xmax1 then xmax1 := x;
                     if y < ymin1 then ymin1 := y;
                     if y > ymax1 then ymax1 := y;
                     if Abs(xmax1 - x) < 1e-5 then
                                begin
Memo1.Clear;
Memo1.Lines.Add('Xmin = ' + FloatToStrF(xmin1,ffFixed,8,5));
Memo1.Lines.Add('Xmax = ' + FloatToStrF(xmax1,ffFixed,8,5));
Memo1.Lines.Add('Ymin = ' + FloatToStrF(ymin1,ffFixed,8,5));
Memo1.Lines.Add('Ymax = ' + FloatToStrF(ymax1,ffFixed,8,5));
Memo1.Lines.Add(' E1 = ' + FloatToStrF(e1,ffFixed,8,5));
                                end;
// Рисование положения планет
                     inc(iv);
                     if iv > 100 then
                     begin
                                     iv := 0;
                                     Chart1.Series[1].AddXY(x,y);
                                     Chart1.Series[2].Clear;
                                     Chart1.Series[2].AddXY(x,y,'',clblue);
                                     Chart1.Series[4].Clear;
                                     Chart1.Series[4].AddXY(x2, y2,",clMaroon);
                                     Chart1.Series[3].AddXY(x2, y2,",clYellow);
                     end;
                       t := t + dt;
                        Application.processmessages;
                                                                                application.processmessages;
                                                                                                                                                                                                               // Pause!!!
                                              repeat
                                                                            not (CheckBox1.Checked);
                                             until
          until checkbox2.checked;
end;
```



#### Результаты выполнения программы:

### Модуль 2.

#### Тема 5. Компьютерное моделирование идеального газа в сосуде.

Создайте новое приложение **Delphi** и разместите на форме следующие элементы



- 1. Button1 (с надписью 1 ШАР), Button2 (N ШАРОВ), Button3 (STOP!!!)
- 2. Chart1
- 3. Checkbox1 с надписью Random, Checkbox2 с надписью STOP!!!
- **4.** Edit1, Edit2, Edit3, Edit4, Edit5. Присвоить значения **5**, **500**, **500**, **2**, **10** соответственно.
- 5. Label1, Label2, Label3, Label4, Label5, Label6 с надписями N=, Xmax=, Ymax=, Vmax=, R =, E = соответственно

На элемент **Chart1** добавить 2 графика, 1 график – line, 2 график – Points Задать следующие свойства кривых:





В Object Tree View выделить Chart1.Series1 и в инспекторе объектов Object Inspector установить для Xvalues.Order значение IoNone Для графика Chart1 убираем галочки в пунктах: Chart1.Title Убираем галочку Visible Chart1.Legend Убираем галочку Visible

Chart1.Axis.Left.Ticks. кнопка "Grid Border..." Убираем гало Chart1.Axis.Bottom.Ticks. кнопка "Grid Border..." Убираем гало

Убираем галочку Visible Убираем галочку Visible

#### Код программы

```
unit Unit1;
              !!! Пропуск интерфейсной части
{$R *.dfm}
procedure TForm1.Button3Click(Sender: TObject);
       CheckBox1.Checked := not (CheckBox1.Checked); end;
begin
                                       _____
//==========
                      1 Молекула
procedure TForm1.Button1Click(Sender: TObject);
var
  x,y, xmin,xmax,ymin,ymax,t,dt,vx,vy,v, x2,y2,vx2,vy2,r,vmax :real;
   i,j,n,ccc : integer;
AB,V1,V2,aPrX,aPrY,aPrXx,aPrXy,aPrYx,aPrYy,bPrXx,bPrXy,bPrX,bPrY,bPrYx,
bPrYy,alfaA,gammaA,alfaB,gammaB,betaA,betaB:Real;
   vs1,vs2,r1,r2,E,Radius:real; itn,Lx,Ly,ix,iy:integer;
begin
                                ymin := 50; ymax := 500;
   xmin := 50; xmax := 500;
   xmax := StrToFloat(Edit2.Text); ymax := StrToFloat(Edit3.Text);
vmax := StrToFloat(Edit4.Text); Radius := StrToFloat(Edit5.Text);
    t := 0; dt := 0.1;
//-----
                       -----
if CheckBox2.Checked then
 begin // Случайные координаты Шара
  x := xmin + (xmax - xmin)*random;
  y := ymin + (ymax - ymin)*random;
                                    end
   else begin // Шар в середине квадрата
   x := (xmin + xmax)/2; y := (ymin + ymax)/2; end;
    r := random*1*pi;
   vx := vmax*cos(r)*random; vy := vmax*sin(r)*random;
//-----
chart1.Series[0].AddXY(xmin,ymin); chart1.Series[0].AddXY(xmax,ymin);
chart1.Series[0].AddXY(xmax,ymax); chart1.Series[0].AddXY(xmin,ymax);
chart1.Series[0].AddXY(xmin,ymin);
chart1.Series[1].Clear;
 checkbox1.Checked := False;
   repeat
                       y := y + vy^*dt;
       x := x + vx^{*}dt;
if x < xmin + 10 then begin x := xmin + 10; vx := -vx; end;
if x > xmax - 10 then begin x := xmax - 10; vx := -vx; end;
if y < ymin + 10 then begin y := ymin + 10; vy := -vy; end;
if y > ymax - 10 then begin y := ymax - 10; vy := -vy; end;
       chart1.Series[1].Clear;
       chart1.Series[1].AddXY(x,y,",clRed);
       t := t + dt;
       application.processmessages;
   until checkbox1.Checked;
end;
```

```
//===========
                             Молекул
                         Ν
                                          =========
procedure TForm1.Button2Click(Sender: TObject);
const
                         eps: real = 1e-5;
    mu : real = 1.0;
var
   xmin,xmax,ymin,ymax,Radius,t,dt,v,dx,dy :real;
   x,y,vx,vy: array [0..1000] of real;
   randomcolor: array [1..3,0..1000] of integer;
   i, i, n, ccc : integer;
                       r,vmax : real;
   AB,V1,V2,aPrX,aPrY,aPrXx,aPrXy,aPrYx,aPrYy,bPrXx,bPrXy,
   bPrX,bPrY,bPrYx,bPrYy,alfaA,gammaA,alfaB,gammaB,betaA,betaB:Real;
   vs1,vs2,r1,r2,E:real;
   itn,Lx,Ly,ix,iy:integer;
begin
    xmin := 0;
                  xmax := 500;
                                     ymin := 0;
                                                   ymax := 500;
                      dt := 0.1;
                                      n := 10;
                                                      itn := 0;
        t := 0;
        n := StrToInt(Edit1.Text);
   xmax := StrToFloat(Edit2.Text);
                                    ymax := StrToFloat(Edit3.Text);
   vmax := StrToFloat(Edit4.Text); Radius := StrToFloat(Edit5.Text);
   if n > 1000 then n := 1000;
    E := 0;
     ix := 1; iy := 1;
     r1 := 4 * Radius;
     Lx := Trunc ((xmax - xmin)/r1) - 0;
     r2 := |x^*|x:
     if r_2 < n then
     begin
         r1 := r1 * r2/n*0.9;
     Lx := Trunc ((xmax - xmin)/r1) - 0;
     end;
for i := 1 to n do
 begin
   if CheckBox2.Checked then
      begin
         x[i] := xmin + (xmax - xmin)*random;
         y[i] := ymin + (ymax - ymin)*random;
      end
        else
      begin
    if (not Odd(iy)) then dx := 0.5*r1 else dx := 0*r1;
    x[i] := xmin + ix*r1 + dx;
    y[i] := ymin + iy*r1;
      inc(ix);
      if ix > = Lx then
        begin
            ix := 1; Inc(iy);
```

```
end;
      end;
//-----
                       _____
    r := random*1*pi;
    vx[i] := vmax*cos(r)*random; vy[i] := vmax*sin(r)*random;
    E := E + sqr(vx[i]) + sqr(vy[i]);
for j := 1 to 3 do randomcolor[j,i] := Random(255);
 end;
chart1.Series[0].Clear;
chart1.Series[0].AddXY(xmin,ymin); chart1.Series[0].AddXY(xmax,ymin);
chart1.Series[0].AddXY(xmax,ymax); chart1.Series[0].AddXY(xmin,ymax);
chart1.Series[0].AddXY(xmin,ymin); chart1.Series[1].Clear;
  checkbox1.Checked := False;
         Label6.Caption := ' E = ' + FloatToStrF(e/n,ffFixed,7,5);
   repeat
    E := 0;
For i := 1 to n do
 begin
        x[i] := x[i] + vx[i]*dt;
                                  y[i] := y[i] + vy[i]*dt;
        if x[i] < xmin + 10 then
                    x[i] := xmin + 10; vx[i] := -vx[i];
        begin
                                                               end;
        if x[i] > xmax - 10 then
                    x[i] := xmax - 10; vx[i] := -vx[i];
        begin
                                                              end;
        if y[i] < ymin + 10 then
                    y[i] := ymin + 10; yv[i] := -vv[i];
        begin
                                                               end;
        if y[i] > ymax - 10 then
        begin
                    y[i] := ymax - 10; yy[i] := -vy[i];
                                                              end;
    E := E + sqr(vx[i]) + sqr(vy[i]);
  end;
         Label6.Caption := ' E = ' + FloatToStrf(e/n,ffFixed,7,5);
 // Проверка сохранения энергии
     V1 := sqrt(vx[i]*vx[i] + vy[i]*vy[i]);
     V2 := sqrt(vx[i]*vx[i] + vy[i]*vy[i]);
    vs2 := sqr(v1) + sqr(v2);
    if vs2 <> vs1 then
    if vs1 > 0 then
    begin
         r := Sqrt(vs1/vs2);
        vx[i] := vx[i] * r;
        vy[i] := vy[i] * r;
        vx[j] := vx[j] * r;
        vy[i] := vy[i] * r;
     end;
     V1 := sqrt(vx[i]*vx[i] + vy[i]*vy[i]);
     V2 := sqrt(vx[j]*vx[j] + vy[j]*vy[j]);
    vs2 := sqr(v1) + sqr(v2);
 // Проверка сохранения энергии
end;
end;
```

end;	
chart1.Series[1].Clear;	
for i := 1 to n do	
chart1.Series[1].AddXY(x[i],y[i],	
",rgb(randomcolor[1,i],randomcolor[2,i],randomcolor[3,i])	);
t := t + dt;	
inc(itn);	
application.processmessages;	
until checkbox1.Checked;	
end;	
//=====================================	
end.	



#### Результаты выполнения программы:

#### Тема 6. Компьютерное моделирование эффекта перколяции.

Создайте новое приложение **Delphi** и разместите на форме следующие элементы



- 7. Button1 (с надписью 1 СТАРТ!), Button2 (с надписью N СТАРТ!).
- 8. Edit1, Edit2, Edit3. Присвоить значения 20, 0.55, 10.
- 9. Label1, Label2, Label3, Label4 с надписями L=, p=, N=, Label4.
- 10. СнескВох1 с надписью "Вывод точек".
- 11. Memo1. Свойство ScrollBar установить ssVertical.
- **1.** Дважды кликнуть по любому пустому месту на форме и набрать **1** блок приведенного ниже листинга программы.
- 2. Дважды кликнуть по кнопке **Button1** и набрать 2 блок программы.
- 3. Дважды кликнуть по кнопке Button2 и набрать 3 блок программы.

#### Код программы

```
unit Unit1;
```

!!! Пропуск интерфейсной части

{\$R \*.dfm}

```
//= 1 ======= При запуске приложения ========
procedure TForm1.FormCreate(Sender: TObject);
begin
   DecimalSeparator := '.';
   Randomize;
   Label4.Caption := ' ';
end;
              Моделирование эффекта перколяции ===
//= 2 ===
procedure TForm1.Button1Click(Sender: TObject);
var
 i,j,k,L,Lc,n,nc,d:Integer;
 x0,y0:Integer;
 p,r,pc:Real;
   F:array [0..201,0..201] of Integer;
beain
    L := StrToInt(Edit1.Text);
   if L > 200 then L := 200;
   Lc := 600;
    d := Trunc(Lc /L);
    p := StrToFloat(Edit2.Text);
   x0 := 50;
   y0 := Lc + 50;
//-----
  for i := 0 to L do for j := 0 to L do F[i,j] := 0;
//-----
      Занимаем узлы с вероятностью р
//
  For i := 1 to L do for j := 1 to L do
   if p > Random then F[i,j] := 1 else F[i,j] := 0;
11
       Рисуем картину
    Canvas.Pen.Color := clWhite;
    Canvas.Brush.Color := clWhite;
    Canvas.Rectangle(x0, y0, x0 + Lc, y0 - Lc);
 if L < 30 then Canvas.Pen.Color := clblack
         else
                Canvas.Pen.Color := clred;
                Canvas.Brush.Color := clred;
  for i := 1 to L do
                    for i := 1 to L do
   if F[i,j] = 1
              then
    Canvas.Rectangle(x0 + i*d-d, y0 - j*d + d, x0 + i*d, y0 - j*d);
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
var
 i,j,k,L,Lc,n,nc,d:Integer; x0,y0:Integer;
 p,r,pc:Real; perk:boolean;
F:array [0..201,0..201] of Integer;
  Pkl:array [0..201,0..201] of Integer;
//-----
function prk(x:boolean):integer;
begin
      if x then prk := 1 else prk := 0; end;
begin
   L := StrToInt(Edit1.Text); N := StrToInt(Edit3.Text);
   if L > 200 then L := 200;
   Lc := 600; d := Trunc(Lc /L);
    p := StrToFloat(Edit2.Text);
   x0 := 50; y0 := Lc + 50; nc := 0;
    memo1.Clear; memo1.Lines.add(' N = ' + FloatToStr(N));
for k := 1 to n do begin
//-----
  for i := 0 to L do for j := 0 to L do
begin F[i,j] := 0; Pkl[i,j] := 0; end;
// Занимаем узлы с вероятностью р
  for i := 1 to L do for j := 1 to L do
     if p > Random then F[i,j] := 1 else F[i,j] := 0;
//----- Определяем кластер -----
     cluster;
//-----
end;
    pc := nc/n;
    if pc > 0.5 then
      begin perk := True;
    Label4.Font.Color := clRed;
       Label4.Caption := ' ПЕРКОЛЯЦИЯ !!! ';
      end
            else
      begin perk := False;
    Label4.Font.Color := clbLACK;
       Label4.Caption := ' НЕТ перколяции ';
      end;
    memo1.Lines.add(' Nc = ' + FloatToStr (Nc));
    memo1.Lines.add(' Pc = ' + FloatToStr (pc));
//
       Рисуем картину
      Canvas.Pen.Color:=clWhite; Canvas.Brush.Color:=clWhite;
      Canvas.Rectangle(x0,y0,x0 + Lc,y0 - Lc);
 if L > 30 then Canvas.Pen.Color:=clred;
                  Canvas.Brush.Color:=clred;
    i:=1; j:=1; Canvas.Brush.Color:=clWhite;
```







#### 5. Образовательные технологии

При изучении дисциплины «Компьютерное моделирование в физике» применяются следующие информационные технологии: активные и интерактивные формы, лекции

, контрольные работы, коллоквиумы, зачеты и экзамены, компьютеры. В течение семестра студенты решают задачи, указанные преподавателем, к каждому семинару. В семестре проводятся контрольные работы (на семинарах). Зачет выставляется после решения всех задач контрольных работ, выполнения домашних и самостоятельных работ.

При проведении занятий используются компьютерные классы, оснащенные современной компьютерной техникой. При изложении теоретического материала используется лекционный зал, оснащенный мультимедиа проекционным оборудованием и интерактивной доской.

По всему лекционному материалу подготовлен конспект лекций в электронной форме и на бумажном носителе, большая часть теоретического материала излагается с применением слайдов (презентаций) в программе **PowerPoint**, а также с использованием интерактивных досок.

Обучающие и контролирующие модули внедрены в учебный процесс и размещены на Образовательном сервере Института физики (<u>http://www.dagphys.ru/</u>) Даггосуниверситета (<u>http://edu.icc.dgu.ru</u>), к которым студенты имеют свободный доступ.

практических занятий используется умение студентов В рамках производить расчеты с помощью средств вычислительной техники. Это позволяет существенно приблизить уровень статистической культуры обработки результатов измерений в практикуме к современным стандартам, принятым в науке и производственной деятельности. На этих занятиях студенты закрепляют навыки, опыт общения с ЭВМ и использования обработки результатов наблюдений, статистических методов что работы специальных совершенно необходимо для В учебных И производственных лабораториях

Для подготовки к практическим (семинарским) занятиям изданы учебно-методические пособия, которые в сочетании с внеаудиторной работой способствуют формированию и развития профессиональных навыков обучающихся.

Электронный учебник. Имеются и используются в учебном процессе электронные учебники по дисциплине «Компьютерное моделирование в физике». Электронный учебник предназначен для самостоятельного изучения теоретического материала курса и построен на гипертекстовой позволяющей работать по индивидуальной образовательной основе, траектории. Гипертекстовая структура позволяет обучающемуся определить не только оптимальную траекторию изучения материала, но и удобный темп работы, и способ изложения материала.

*Компьютерная тестирующая система*. Разработана и внедрена в учебный процесс компьютерная тестирующая система, которая

обеспечивает, с одной стороны, возможность самоконтроля для обучаемого, а с другой стороны используется для текущего или итогового контроля знаний студентов.

*Презентация.* Разработан электронный курс лекций по всем темам, с использованием электронных презентаций. Что улучшает восприятие материала, повышает мотивацию познавательной деятельности и способствует творческому характеру обучения.

Учебно-исследовательская работа. В процессе изучения дисциплины используется данная форма практической самостоятельной работы студента, позволяющая студентам изучать научно-техническую информацию ПО процессы, заданной теме. моделировать проводить расчеты по разработанному алгоритму с применением ЭВМ и сертифицированного программного обеспечения, участвовать в экспериментах, анализировать и обрабатывать Результаты исследований полученные результаты. представляются на научно-практических конференциях.

Для усвоения дисциплины используются электронные базы учебнометодических ресурсов, электронные библиотеки.

Удельный вес занятий, проводимых в интерактивных формах, с использованием современных компьютерных средств обучения и демонстрации в учебном процессе составляет не менее 40% лекционных занятий.

#### 6. Учебно-методическое обеспечение самостоятельной работы студентов.

В течение семестра студенты выполняют:

- домашние задания, выполнение которых контролируется и при необходимости обсуждается на лабораторных занятиях;
- промежуточные контрольные работы во время лабораторных занятий для выявления степени усвоения пройденного материала;
- выполнение итоговой контрольной работы по лабораторным занятиям, охватывающие базовые вопросы курса: в конце семестра.

*Итоговый контроль: зачет* в конце семестра, включающий проверку теоретических знаний и умение решения по всему пройденному материалу.

## 6.1 Учебно-методическое обеспечение самостоятельной работы студентов.

#### Виды и порядок выполнения самостоятельной работы:

- 1. Изучение рекомендованной литературы
- 2. Поиск в Интернете дополнительного материала
- 3. Подготовка реферата (до 5 страниц), презентации и доклада (10-15 минут)
- 4. Подготовка к зачету.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов:

#### Виды и порядок выполнения самостоятельной работы:

№ п/п	Вид самостоятельной работы	Вид контроля	Учебно-методич. обеспечение
1.	Подготовка реферата (до	Прием реферата,	См. разделы 6.1, 6.2
	5 страниц), презентации	презентации, доклада и	и 7 данного
	и доклада (10-15 минут)	оценка качества их	документа
		исполнения на мини-	
		конференции.	
2.	Подготовка к зачету	Промежуточная аттестация в	См. разделы 6.3, 6.4
		форме зачета.	и 7 данного
			документа

- **1.** Текущий контроль: Прием реферата, презентации, доклада и оценка качества их исполнения на мини-конференции.
- 2. Промежуточная аттестация в форме зачета.

**Текущий контроль** успеваемости осуществляется непрерывно, на протяжении всего курса. Прежде всего, это устный опрос по ходу лекции, выполняемый для оперативной активизации внимания студентов и оценки их уровня восприятия. Результаты устного опроса учитываются при выборе экзаменационного вопроса. Примерно с пятой недели семестра - в форме контроля самостоятельной работы по подготовке рефератов, содержание которых будет представлено публично на мини-конференции и сопровождено презентацией и небольшими тезисами в электронной форме.

Выбор темы реферата согласуется с лектором.

Практикуется два типа тем - самостоятельное изучение конкретной проблемы или ознакомление с учебным дистанционным курсом по теме курса. Результаты самостоятельной работы играют роль допуска к экзамену.

#### Промежуточная аттестация:

Для допуска к зачету надлежит сделать сообщение на миниконференции, представить презентацию и собственно текст реферата. Зачет проходит в устной форме в виде ответов на билеты и, если понадобится, то на дополнительные контрольные вопросы, которые задает экзаме-

натор при необходимости уточнить оценку.

- Оценка «отлично» ставится за уверенное владение материалом курса и демонстрацию способности самостоятельно анализировать вопросы применения и развития современных ИТ.
- Оценка «хорошо» ставится при полном выполнении требований к прохождению курса и умении ориентироваться в изученном материале.
- Оценка «удовлетворительно» ставится при достаточном выполнении требований к прохождению курса и владении конкретными знаниями по программе курса.
- Оценка «неудовлетворительно» ставится, если требования к прохождению курса не выполнены и студент не может показать владение материалом курса.

## 6.2 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Тематика рефератов ежегодно подвергается пересмотру и обновлению соответственно появлению новых перспективных средств и методов работы с информацией. Предлагается следующий список рефератов, который может быть расширен и уточнен при обсуждении и конкретизации с студентами.

#### 6.3. Примеры тем рефератов.

- Метод наименьших квадратов.
- Интерполяционный многочлен Лагранжа.
- Численное интегрирование методом Монте-Карло.
- Методы Рунге-Кутта для решения дифференциальных уравнений.
- Решение дифференциальных уравнений второго порядка методом Рунге-Кутта.
- Методы математического моделирования решения нелинейных уравнений.
- Минимизация функций многих переменных. Современные методы.
- Классификация математических моделей. Примеры моделей.
- Классический метод Монте-Карло
- Современные алгоритмы метода Монте-Карло
- Моделирование движения тела в гравитационном поле.
- Моделирование движения планет солнечной системы.
- Моделирование броуновского движения частиц.
- Моделирование систем с перколяцией.
- Моделирование отражения света на границе двух сред.
- Моделирование распространения света через среду.
- Исследование модели Изинга методом Монте-Карло

#### 6.4. Рекомендации к последовательности выполнения реферата.

А) Изучение проблемы по материалам, доступным в Интернете:

- 1. Согласовать название сообщения.
- 2. Написать тезисы реферата по теме.
- 3. Выразить, чем интересна выбранная тема в наши дни.
- 4. Подготовить презентацию по выбранной теме.
- 5. Сделать сообщение на мини-конференции.
- Б) Ознакомление с заданным дистанционным курсом:
  - 1. Представить основные идеи заданного курса.
  - 2. Описать достоинства и недостатки материала, изложенного в данном курсе.
  - 3. Написать отзыв на данный курс.
  - 4. Сформулировать рекомендации по применению данного курса.
  - 5. Сделать сообщение о содержании курса на мини-конференции.

## 7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

Перечень компетенций с указанием этапов их формирования приведен в описании образовательной программы.

Компетенция	Знания, умения, навыки	Процедура освоения
ОПК-5	Умение: использовать в	Письменный опрос
	профессиональной деятельности	
	базовые знания фундаментальных	
	разделов математики, создавать	
	математические модели типовых	
	профессиональных задач и	
	интерпретировать полученные	
	результаты с учетом границ	
	применимости моделей; использовать	
	базовые теоретические знания	
	фундаментальных разделов общей и	
	теоретической физики для решения	
	профессиональных задач	
	Умение проводить научные	
	исследования в избранной области	
	экспериментальных и (или)	
	теоретических физических	
	исследований с помощью современной	
	приоорнои оазы (в том числе сложного	
	физического оборудования) и	
	информационных технологии с учетом	
	отечественного и	
	заруоежного опыта	
Πν 5		V numeri v oroz
IIK-J	области физики лид осрозния	круглый стол
	профильных физика для освоения	
	профильтыл физических дисциплин,	
	анализа и синтеза физической	
	информации в избранной области	
	Ангрормации в изоранной области	
	информации в избранной области физических исследований	

#### 7.2. Типовые контрольные задания

- 1. Компьютерное моделирование в физике. Постановка задачи. Корректность задачи.
- 2. Погрешности вычислений. Абсолютная и относительная погрешность.
- 3. Приближение функций. Интерполяция функций. Подбор эмпирических формул.

- 4. Квадратичная интерполяция.
- 5. Интерполяционный многочлен Лагранжа.
- 6. Аппроксимация функций. Метод наименьших квадратов.
- 7. Методы численного интегрирования. Метод прямоугольников.
- 8. Метод трапеций. Метод Симпсона.
- 9. Методы численного интегрирования. Метод Монте-Карло.
- 10. Методы математического моделирования решения обыкновенных дифференциальных уравнений.
- 11. Решение дифференциальных уравнений второго порядка.
- 12. Методы Рунге-Кутта для решения дифференциальных уравнений.
- 13. Методы математического моделирования решения нелинейных уравнений.
- 14. Метод Ньютона (касательных). Метод секущих.
- 15. Методы математического моделирования минимизации. Нахождение экстремумов функций.
- 16. Нахождение экстремумов функций. Метод золотого сечения.
- 17. Методы математического моделирования минимизации функций многих переменных. Симплексные методы.
- 18. Симплексный метод Нелдера-Мида.

# 7.3. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Текущий контроль осуществляется в виде опросов (0-1) и теста (0-1-2) по окончании каждого раздела программы и коллоквиума (1-2) по окончании курса.

Активное участие в предоставлении (2-3-4 балла) и обсуждении докладов (1-2), групповой работе (1-2) и успешное участие в опросах (1-2) обеспечивают накопление баллов. Удачная сдача теста, активное участие в коллоквиумах, в совокупности с нормальной текущей работой, означают автоматическое получение допуска к дифференцированному зачету.

Пропуски и опоздания на занятия означают вычет соответственно 2 и 1 балла из накопленной суммы. Тем, кто не набрал достаточного количества баллов, дается дополнительное домашнее задание в виде реферата (0-1-2-3-4-5 баллов). Оценка по итогам работы на семинарах в течение учебного года определяется суммой баллов: менее 0 баллов – оценка «2», от 1 до 9 – «3», от 10 до 14 – «4», более 14 – «5».

## 8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.

#### а) основная литература:

- 1. Нартя В.И. Блочно-матричный метод математического моделирования поверхностей [Электронный ресурс] / В.И. Нартя. Электрон. текстовые данные. М. : Инфра-Инженерия, 2016. 236 с. 978-5-9729-0119-7. Режим доступа: <u>http://www.iprbookshop.ru/51718.html</u>
- шумового воздействия транспорта 2. Оценка уровня методом математического моделирования (расчетный метод) [Электронный ресурс]: методические указания к выполнению практических работ по дисциплине «Проектирование и реконструкция зданий» для студентов магистратуры направления подготовки 08.04.01 Строительство / . — Электрон. текстовые данные. — М. : Московский государственный строительный университет, Ай Пи Эр Медиа, ЭБС АСВ, 2015. — 32 с. 978-5-7264-1096-8. Режим доступа: http://www.iprbookshop.ru/36149.html
- 3. Пушкарева А.Е. Методы математического моделирования в оптике биоткани [Электронный ресурс] : учебное пособие / А.Е. Пушкарева. Электрон. текстовые данные. СПб. : Университет ИТМО, 2008. 103 с. 2227-8397. Режим доступа: http://www.iprbookshop.ru/67285.html

#### б) дополнительная литература:

- 1. Магомедов М.А., Муртазаев А.К., Хизриев К.Ш. Методы математического моделирования. Учебно-методическое пособие. Махачкала: 2007. 50с.
- Муртазаев А.К., Магомедов Г.М., Рамазанов М.К., Магомедов М.А., Методы численного эксперимента в физике. Учебное пособие. – Махачкала: 2009. – 58с.
- 3. К. Биндер, Д.В. Хеерман. Моделирование методом Монте-Карло в статистической физике. М. 1995г.
- 4. Д.В. Хеерман. Методы компьютерного эксперимента в теоретической физике. М. 1990 г.
- 5. С. Кунин. Вычислительная физика М. 1992 г.
- 6. Рябенький В.С. Введение в вычислительную математику. М/: Наука-Физматлит, 1994. – 335с. 2-е изд. М: Физматлит, 2000. – 296 с.
- 7. Федоренко Р.П. Введение в вычислительную физику. М.: Изд-во МФТИ, 1994. – 528 с.
- 8. Киреев В.И., Пантелеев А.В. Методы математического моделирования в примерах и задачах. М.: Изд-во МАИ, 2000.
- 9. Каханер Д., Моулер К., Нэш С. Методы математического моделирования и программное обеспечение. М.: Мир, 1998. 575с.

10.Бахвалов Н.С., Жидков Н.П., Кобельков Г.Г. Численные методы. 8-е изд. – М.: Лаборатория Базовых Знаний, 2000. – 624с.

 Лобанов А.И., Петров И.Б. Вычислительные методы для анализа моделей сложных динамических систем. Часть 1. – М.: МФТИ, 2000. – 168с.

## 9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.

- 1. ЭБС IPRbooks: <u>http://www.iprbookshop.ru/</u> Лицензионный договор № 2693/17от 02.10.2017г. об оказании услуг по предоставлению доступа. Доступ открыт с с 02.10.2017 г. до 02.10.2018 по подписке
- 2. Электронно-библиотечная система «Университетская библиотека онлайн» <u>www.biblioclub.ru</u> договор № 55\_02/16 от 30.03.2016 г. об оказании информационных услуг.(доступ продлен до сентября 2019 года).
- 3. Доступ к электронной библиотеки на <u>http://elibrary.ru</u> основании лицензионного соглашения между ФГБОУ ВПО ДГУ и «ООО» «Научная Электронная библиотека» от 15.10.2003. (Раз в 5 лет обновляется лицензионное соглашение)
- 4. Национальная электронная библиотека <u>https://нэб.рф/</u>. Договор №101/НЭБ/101/НЭБ/1597
- 5. Федеральный портал «Российское образование» <u>http://www.edu.ru/</u> (единое окно доступа к образовательным ресурсам).
- 6. Федеральное хранилище «Единая коллекция цифровых образовательных ресурсов» <u>http://school-collection.edu.ru/</u>
- 7. Российский портал «Открытого образования» <u>http://www.openet.edu.ru</u>
- 8. Сайт образовательных ресурсов Даггосуниверситета <u>http://edu.icc.dgu.ru</u>
- 9. Информационные ресурсы научной библиотеки Даггосуниверситета <u>http://elib.dgu.ru</u> (доступ через платформу Научной электронной библиотеки elibrary.ru).
- 10.Федеральный центр образовательного законодательства <u>http://www.lexed.ru</u>

## 10. Методические указания для обучающихся по освоению дисциплины.

Самостоятельная работа студентов реализуется в виде:

- подготовки к контрольным работам;
- подготовки к семинарским (практическим) занятиям;
- выполнения индивидуальных заданий по основным темам дисциплины;
- написание рефератов по проблемам дисциплины "Физика фазовых переходов и критических явлений".

- обязательное посещение лекций ведущего преподавателя;
- лекции основное методическое руководство при изучении дисциплины, наиболее оптимальным образом структурированное и скорректированное на современный материал;
- в лекции глубоко и подробно, аргументировано и методологически строго рассматриваются главные проблемы темы;
- в лекции даются необходимые разные подходы к исследуемым проблемам;
- подготовку и активную работу на лабораторных занятиях;
- подготовка к лабораторным занятиям включает проработку материалов лекций, рекомендованной учебной литературы.

#### 11.Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.

Образовательные технологии, применяемые на практических и лабораторных занятиях:

- 1. Технология активного (контекстного) обучения (коллективная работа малыми группами исследовательская игра: группа разбивается на подгруппы, в каждой из которых назначается руководитель (определяет цели и задачи, назначает ответственных за отдельные задачи, координирует работу и представляет общее решение задачи) и исполнители (решают отдельные задачи);
- 2. Технология деловой игры (имитационная соревновательная игра: малые группы получают одинаковое задание, распределяются по ролям (руководитель, ответственные исполнители) и выполняют его на скорость и качество, которое оценивается преподавателем);
- 3. Технология интерактивного обучения (мозговой штурм: группа получает задание, далее предлагается высказывать как можно большее количество вариантов решения, затем из общего числа высказанных идей отбирают наиболее удачные, которые могут быть использованы на практике).

## 12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.

Для материально-технического обеспечения дисциплины «Компьютерное моделирование в физике» используется:

• компьютерные классы физического факультета.