

моделирования экспериментальных спектров для температур ниже 171 К в виде их линейной комбинации.

Литература

1. Coussan S., Bakkas N., Loutellier A., Perchard J.P., Racine S. Infra-red photoisomerization of the methanol cyclic trimer trapped in a nitrogen matrix / S. Coussan et al // Chem. Phys. Lett. – 1994 – Vol. 217, No. 1, 2 – P. 123–130.

2. Schuster P. The Hydrogen Bond – Recent Developments in Theory and Experiment / P. Schuster, G. Zundel, C. Sandorfy // North Holland Publ. Co., Amsterdam – 1976.

Е.М. Кончиц (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **Е.А. Дей**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПОСТРОЕНИЯ ГРАФИКОВ ЭЛЕМЕНТАРНЫХ ФУНКЦИЙ В СИСТЕМЕ ANDROID

Система Android – одна из самых молодых и перспективных мобильных операционных систем (ОС), основанная на базе операционной системы Linux и разрабатываемая Open Handset Alliance (ОНА) при поддержке Google. К достоинствам этой ОС можно отнести гибкость, открытые исходные коды, большое количество готовых программ, высокое быстродействие, удобное взаимодействие с сервисами от Google, многозадачность. К недостаткам можно отнести множество актуальных версий и невысокую защищённость от хакерских атак из-за открытости кода.

Графику в android приложении можно создавать двумя способами: нарисовав графику в объекте view из компоновки, а также нарисовав графику непосредственно на канве. Android SDK предлагает двумерную графическую библиотеку рисования на формах и изображениях – android.graphics.drawable. Класс Drawable является базовым классом для всех классов работы с графикой. Это общая абстракция для рисуемого объекта. Всего существует два способа реализации рисования на канве: в основном потоке программы, в котором запускается Activity, создаётся собственный компонент view, затем вызывается метод invalidate() и обрабатывается создание графики в методе обратного вызова onDraw(), и в отдельном потоке – через объект SurfaceView. Класс canvas имеет собственный набор методов для рисования, которые можно использовать, например drawPoint() – этот метод рисует точку,

`drawLine()` – этот метод рисует линию, между двумя точками, `drawText()` – данный метод рисует текст. Канва фактически является поверхностью, на которой графика будет рисоваться. Когда выполняется прорисовка в пределах метода обратного вызова `view.onDraw()`, система передает в качестве параметра объект `canvas`. Система Android вызывает метод `onDraw()` по мере необходимости. Каждый раз, когда изображение на канве представления требует перерисовки, необходимо вызывать метод `invalidate()`. Он требует от системы обновления представления, и система Android тогда вызовет метод `onDraw()`.

Для того чтобы более подробно изучить графику в операционной системе (ОС) Android, мною было создано приложение, отображающее графики элементарных функций. В качестве среды разработки была выбрана программа Android Studio. Android Studio – IDE для разработки приложений для Android от Google на основе IntelliJ IDEA.

Разработанное приложение представляет собой окно, на котором изображается график функции. В приложении присутствует выпадающий список, представленный элементарными функциями. По нажатию любого элемента из списка, приложение изображает график выбранной пользователем функции.

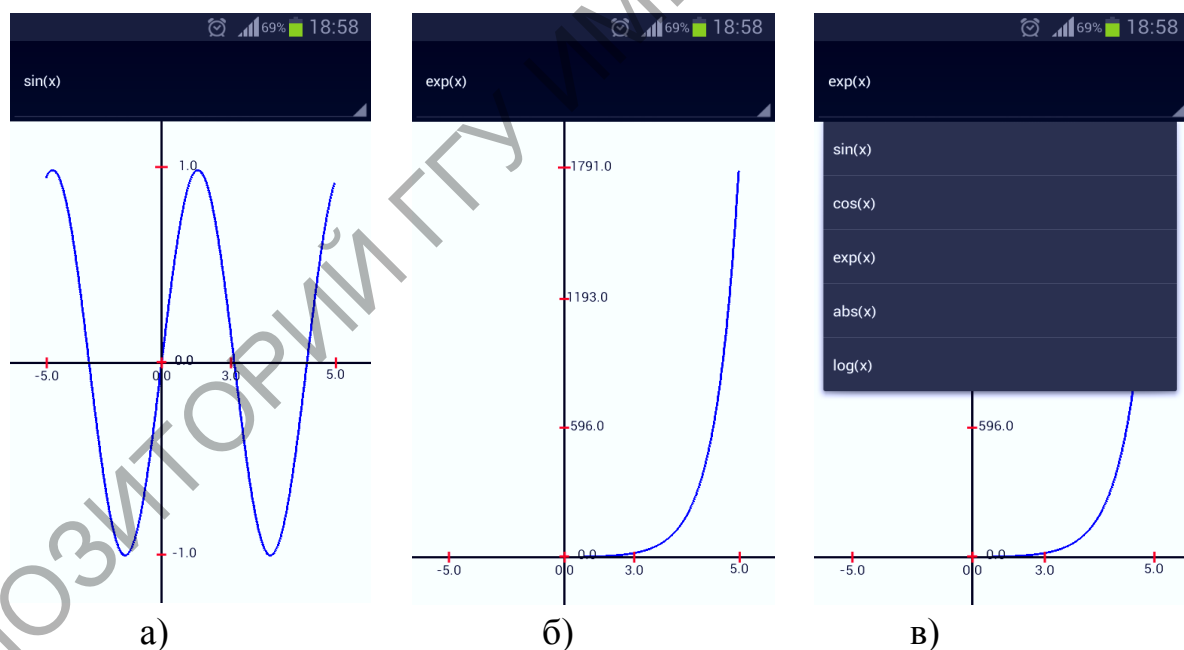


Рисунок 1 – Вид экрана: а) – при отображении функции « $\sin(x)$ », б) – при отображении функции « $\exp(x)$ », в) – при отображении списка доступных функций

Приложение состоит из двух классов. Первый класс `MainActivity.java` содержит в себе ссылку на корневой элемент представления (`layout-файл`), который используется для графической прорисовки

изображения на экране мобильного телефона, а также методы для создания списка функций. Второй класс PlotView.java – это наследник класса View. В данном классе происходит рисование самого графика функции, а также осей координат. Для того, чтобы нарисовать линию графика и оси координат используется метод drawLine(). Для отображения чисел на осях координат используется метод drawText().

Пример организации рисования линии графика в коде программы:

```
paint.setStrokeWidth(3);  
paint.setColor(Color.BLUE);  
canvas.drawLine(xvalues[i],canvasHeight-  
yvalues[i],xvalues[i+1],canvasHeight-yvalues[i+1],paint);
```

Методы paint.setStrokeWidth(3) и paint.setColor(Color.BLUE) отвечают за толщину и цвет линии графика соответственно.

Разработанное приложение может быть доработано, для дальнейшего использования его в решении физических задач.

Литература

1. Голощапов, А.Л. Google Android. Создание приложений для смартфонов и планшетных ПК/ А.Л. Голощапов. – СПб.: БХВ-Петербург, 2013. – 832 с.: ил. – (В подлиннике).

2. Дэрси, Л. Android за 24 часа. Программирование приложений под операционную систему Google / Л. Дэрси, Ш. Кондор – М.: Рид Групп, 2011. – 464 с. – (Профессиональные компьютерные книги).

Ю.В. Кореновская (БГУ, Минск)

Науч. рук. **И.В. Сташкевич**, канд. физ.-мат. наук, доцент

ВЛИЯНИЕ ТЕМПЕРАТУРЫ НА ГЕНЕРАЦИОННЫЕ ХАРАКТЕРИСТИКИ НЕОДИМОВОГО ЛАЗЕРА С КВАЗИ ТРЕХУРОВНЕВОЙ СХЕМОЙ

В настоящее время наиболее перспективным методом получения высококогерентного излучения в синем диапазоне длин волн является преобразование во вторую гармонику излучения неодимового лазера, работающего по квази трехуровневой схеме (рисунок 1).

В этом случае генерация происходит на длине волны 946 нм на переходах ${}^4F_{3/2} - {}^4I_{9/2}$, причем нижним лазерным уровнем является самый верхний подуровень расщепленного основного состояния. Его