**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC HOA SEN**

**KHOA KHOA HỌC VÀ CÔNG NGHỆ**

**ĐỒ ÁN THỰC TẬP LẬP TRÌNH B**

**TÊN ĐỀ TÀI:**

**PHẦN MỀM MÁY TÍNH TRÊN ANDROID**

* Giảng viên hướng dẫn: Cô Nguyễn Thị Thanh

# Lớp: QL101A

* **Nhóm :**

1. Cao PhướcThanh
2. Phạm Đức Luân
3. Huỳnh Minh Hoàng
4. Hoàng Ngọc Tân

**THÁNG 7 NĂM 2012**

*Nhận xét của giảng viên hướng dẫn*

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………  
***Điểm cá nhân Điểm Nhóm***

1.

2.

3.

4.

5.

Ngày …. tháng …. năm 2012

Giảng viên

TRÍCH YẾU

*Do yêu cầu của đề án thực tập B nhóm thực hiện đồ án cũng như thực hiện phần mềm này nhằm nâng cao giá trị sử dụng cho smartphone với phần mềm máy tính khoa học. Phần mềm được xây dựng trên nền tảng Java tích hợp vào hệ điều hành Adroid 4.0.3.*

*Khi xây dựng đồ án đòi hỏi chúng tôi phải tìm hiểu nhưng kiế thức mới, như việc sử dụng ngôn ngữ lập trình Java đồng thời xem lại các công thức toán học đã được học. Ngoài ra trong quá trình thực hiện đồ án, nhóm đã hiểu hơn về lập trình ứng dụng cho smartphone, ngôn ngữ Java,công thức toán học, cách sử dụng chương trình lập trình Eclipse SDK và có thêm nhiều kinh nghiệm làm việc theo nhóm.*

*Xây dựng đồ án tuy gặp rất nhiều khó khăn nhưng nhóm đã cố gắng vượt qua và hoàn thành đồ án một cách tốt nhất và hoàn thành đúng thời hạn các giai đoạn.*

# MỤC LỤC

[1](#_Toc329250186)

[TRÍCH YẾU 3](#_Toc329250187)

[MỤC LỤC 4](#_Toc329250188)

[LỜI CẢM ƠN 6](#_Toc329250189)

[NHẬP ĐỀ 7](#_Toc329250190)

[GIỚI THIỆU 8](#_Toc329250191)

[I. Đôi nét về ngôn ngữ Java: 8](#_Toc329250192)

[II. Hệ điều hành Android: 9](#_Toc329250193)

[III. Định nghĩa thuật ngữ chuyên dùng 10](#_Toc329250194)

[ĐỀ TÀI 11](#_Toc329250195)

[I. MÔ TẢ YÊU CẦU ĐỀ TÀI 11](#_Toc329250196)

[II. YÊU CẦU ĐỒ ÁN VÀ THANG ĐIỂM 12](#_Toc329250197)

[NỘI DUNG CÁO BÁO 15](#_Toc329250198)

[I. DỰ KIẾN BAN ĐẦU: 15](#_Toc329250199)

[II. QUÁ TRÌNH PHÂN CÔNG THỰC HIỆN: 16](#_Toc329250200)

[III. CHỨC NĂNG PHẦN MỀM: 16](#_Toc329250201)

[IV. HƯỚNG DẪN SỮ DỤNG 18](#_Toc329250202)

[1. Tổng quát 18](#_Toc329250203)

[2. Chi tiết 19](#_Toc329250204)

[3. Các nút chuyển đổi 19](#_Toc329250205)

[3.1 Inv 19](#_Toc329250206)

[3.2 Chuyển đổi góc 19](#_Toc329250207)

[4. Nhập biểu thức: 20](#_Toc329250208)

[5. Phép tính với hàm 20](#_Toc329250209)

[5.1 Hàm lượng giác 20](#_Toc329250210)

[5.2 Hàm căn bậc 2 20](#_Toc329250211)

[5.3 Hàm căn bậc n 20](#_Toc329250212)

[5.4 Hàm mũ 21](#_Toc329250213)

[5.5 Hàm logarit 21](#_Toc329250214)

[6. Một số nút chức năng khác 21](#_Toc329250215)

[V. MÔ TẢ PHẦN MỀM: 21](#_Toc329250216)

[VI. Ý TƯỞNG CODE, ÁP DỤNG CÁC CÔNG THỨC VÀ THUẬT TOÁN 22](#_Toc329250217)

[1. CalculateActivity.java : 22](#_Toc329250218)

[2. FixedPoint.java : 25](#_Toc329250219)

[3. Balan.java: 26](#_Toc329250220)

[4. Math.java: 29](#_Toc329250221)

[6. Help.java 37](#_Toc329250222)

[GIAO DIỆN 38](#_Toc329250223)

[I. Giao diện biểu thức: 38](#_Toc329250224)

[II. Giao diện giúp đỡ khi chon button Help 38](#_Toc329250225)

[III. Giao diện thay đổi khi chọn button Inv: 39](#_Toc329250226)

[KẾT QUẢ VÀ ĐÁNH GIÁ KẾT QUẢ 39](#_Toc329250227)

[KẾT LUẬN 40](#_Toc329250228)

[I. Bài học kinh nghiệm: 40](#_Toc329250229)

[II. Hướng phát triển: 40](#_Toc329250230)

[TÀI LIỆU THAM KHẢO 41](#_Toc329250231)

LỜI CẢM ƠN

*Để báo cáo này hoàn thành tốt đẹp, chúng tôi xin gửi lời cảm ơn chân thành đến cô Nguyễn Thị Thanh đã hướng dẫn và sẵn sàng giải đáp những thắc mắc của chúng tôi trong suốt thời gian thực hiện cho đến khi hoàn thành đề án này. Nhờ sự giúp đỡ nhiệt tình của cô, chúng tôi đã có thể hoàn thành đề án một cách tốt nhất. Cám ơn sư hướng dẫn của cô vì đã trang bị tài liệu, kiến thức, kĩ năng, phương pháp và từng bước thực hiện cho nhóm nhằm hạn chế và khắc phục các sai sót trong quá trình thực hiện đề án.*

*Nhóm chúng tôi chúc cô nhiều sức khỏe và ngày càng thành công*

*Cuối cúng, xin cảm ơn những thành viên trong nhóm, những người đã rất nhiệt tình trong quá trình làm việc nhóm để hoàn thành đề án này.*

*Trân trọng*

NHẬP ĐỀ

*Ngày nay, smartphone hẳn không còn xa lại đối với mọi người. Chính vì vậy, việc sử dụng 1 máy tính đơn giản không còn đáp ứng đủ yêu câu về công việc cũng như tiện ích cho người sử dụng như trước đây. Do đó việc xây dựng 1 phần mềm đáp ứng được những nhu cầu về công việc cũng như tiện ích trên điện thoại trở nên cần thiết.*

*Thông qua đồ án lập trình A. Nhóm đã thực hiên, xây dựng được 1 phần mềm máy tính hoàn chỉnh và nâng cao được khả năng lập trình cũng như việc tìm hiểu những ngôn ngữ lập trình, những thuật toán và những khái niệm ứng dụng hoàn toàn mới.*

GIỚI THIỆU

1. Đôi nét về ngôn ngữ Java:

**Java** được khởi đầu bởi [James Gosling](http://vi.wikipedia.org/wiki/James_Gosling) và bạn đồng nghiệp ở [Sunicrosystems](http://vi.wikipedia.org/wiki/Sun_Microsystems) năm [1991](http://vi.wikipedia.org/wiki/1991). Ban đầu ngôn ngữ này được gọi là Oak (có nghĩa là [cây sồi](http://vi.wikipedia.org/w/index.php?title=C%C3%A2y_s%E1%BB%93i&action=edit&redlink=1); do bên ngoài cơ quan của ông Gosling có trồng nhiều loại cây này), họ dự định ngôn ngữ đó thay cho [C++](http://vi.wikipedia.org/wiki/C%2B%2B), nhưng các tính năng giống [Objective C](http://vi.wikipedia.org/w/index.php?title=Objective_C&action=edit&redlink=1). Không nên lẫn lộn Java với [JavaScript](http://vi.wikipedia.org/wiki/JavaScript), hai ngôn ngữ đó chỉ giống tên và loại [cú pháp](http://vi.wikipedia.org/wiki/C%C3%BA_ph%C3%A1p) như [C](http://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF)). Công ty Sun Microsystems đang giữ bản quyền và phát triển Java thường xuyên. Tháng 04/2011, công ti Sun Microsystems tiếp tục cho ra bản JDK 1.6.24.

Java là một [ngôn ngữ lập trình](http://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) dạng [lập trình hướng đối tượng](http://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng) (OOP). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì [biên dịch](http://vi.wikipedia.org/wiki/Tr%C3%ACnh_bi%C3%AAn_d%E1%BB%8Bch) [mã nguồn](http://vi.wikipedia.org/wiki/M%C3%A3_ngu%E1%BB%93n) thành [mã máy](http://vi.wikipedia.org/wiki/M%C3%A3_m%C3%A1y) hoặc [thông dịch](http://vi.wikipedia.org/wiki/Tr%C3%ACnh_th%C3%B4ng_d%E1%BB%8Bch) mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành [bytecode](http://vi.wikipedia.org/w/index.php?title=Bytecode&action=edit&redlink=1), bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy. Bằng cách này, Java thường chạy chậm hơn những ngôn ngữ lập trình thông dịch khác như [C++](http://vi.wikipedia.org/wiki/C%2B%2B), [Python](http://vi.wikipedia.org/wiki/Python), [Perl](http://vi.wikipedia.org/wiki/Perl), [PHP](http://vi.wikipedia.org/wiki/PHP), [C#](http://vi.wikipedia.org/wiki/C)...

[Cú pháp](http://vi.wikipedia.org/wiki/C%C3%BA_ph%C3%A1p) Java được vay mượn nhiều từ [C](http://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) & [C++](http://vi.wikipedia.org/wiki/C%2B%2B) nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn.

Cơ chế hoạt động: ban đầu Sun nghĩ ra một cỗ máy, đặt tên là JVM. Cỗ máy này vẫn được Sun sản xuất, cho tới nay đã được 4.5 tỉ đơn vị.Nói Java compiler không compile ra machine code là không đúng. Thực chất file code Java sau khi đưa vào compiler sẽ được compile ra thành machine code dành cho cái máy này. Trên các platform thực, JRE sẽ emulate JVM để recompile toàn bộ bytecode thành native code của platform đó. Để cho các chương trình Java chạy trên nhiều platform khác nhau, Sun chỉ việc port cái emulator, tức JRE, sang các platform đó. Nhờ vậy mà một chương trình Java đồ sộ viết cho máy tính, đem sang điện thoại di động vẫn chạy được bình thường.

1. Hệ điều hành Android:

**Khái niệm**: Android là hệ điều hành trên [điện thoại di động](http://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_tho%E1%BA%A1i_di_%C4%91%E1%BB%99ng) (và hiện nay là cả trên một số [đầu phát HD](http://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BA%A7u_ph%C3%A1t_HD&action=edit&redlink=1), HD Player) phát triển bởi [Google](http://vi.wikipedia.org/wiki/Google) và dựa trên nền tảng [Linux](http://vi.wikipedia.org/wiki/Linux). Trước đây, Android được phát triển bởi công ty liên hợp Android ( sau đó được Google mua lại vào năm 2005). Theo NPD, thiết bị di động sử dụng hệ điều hành android bán được tại Mỹ trong quý II năm 2010 xếp vị trí đầu tiên với 33%, thứ 2 là BB os với 28% và iOS ở vị trí thứ 3 với 22%. Android có một cộng đồng những nhà phát triển rất lớn viết các ứng dụng cho hệ điều hành của mình.Hiện tại có khoảng 70,000 ứng dụng cho Android os và vào khoảng 100,000 ứng dụng đã được đệ trình, điều này khiến Android trở thành hệ điều hành di động có môi trường phát triển lớn thứ 2. Các nhà phát triển viết ứng dụng cho Android dựa trên ngôn ngữ Java. Sự ra mắt của Android vào ngày 5 tháng 11 năm 2007 gắn với sự thành lập của liên minh thiết bị cầm tay mã nguồn mở, bao gồm 78 công ty phần cứng, phần mềm và viễn thông nhằm mục đính tạo nên một chuẩn mở cho điện thoại di động trong tương lai. Google công bố hầu hết các mã nguồn của Android theo bản cấp phép Apache.

**Đặc tính thông dụng**: Tái hiện lại chương trinh của một máy vi tính trên một smartphone. Ngày nay nhu cầu của con người càng tăng cao cho nên yêu cầu công nghệ cũng phai phát triển để đáp ứng được nhu cầu đó, và sau đây la một số thông dụng chính: nếu chúng ta đi công tác mà không mang theo máy tính hoặc chổ chúng ta công tác chưa có máy vi tính thì android sẽ là một công cụ hữu hiệu và thiết thực nhất. trong hệ điều hành này có sẵn các ứng dụng phục vụ cho nhu cầu người dung nhưng những công cụ này cũng chỉ đáp ứng được phần nào nhu cầu của người dùng, chính vì vậy cần phải mở rộng và phát triển thêm.

1. Định nghĩa thuật ngữ chuyên dùng

**Scientific Calculator:** Máy Tính Khoa Học, dùng để tính toán.

**Smart Phone:** Điện Thoại Cầm Tay Thông Minh, Smartphone ưu việt hơn điện thoại thông thường ở chỗ nó được tích hợp thêm các tính năng văn phòng hỗ trợ người dùng. Ngoài ra, nếu so với PDA phone thì smartphone lại có tính ổn định về chức năng thoại và kết nối mạng tốt hơn.

ĐỀ TÀI

*Xây Dựng Phần Mềm Ứng Dụng Máy Tính Khoa Học (Scientific Calculator) cho Điện Thoại Cầm Tay Thông Minh (Smart Phone) chạy trên hệ điều hành Android.*

1. MÔ TẢ YÊU CẦU ĐỀ TÀI

Gần đây điện thoại cầm tay thông minh đã trở thành một vật hầu như “cần phải có” trong cuộc sống hằng ngày của giới làm việc “trí thức”. Viết phần mềm tiện ích nâng cao giá trị sử dụng cho chúng cũng trở nên một trào lưu cho các công ty phát triển phần mềm, các giới lập trình viên chuyên nghiệp cũng như không chuyên. Đặc biệt là những ứng dụng cho máy chạy trên hệ điều hành Android.

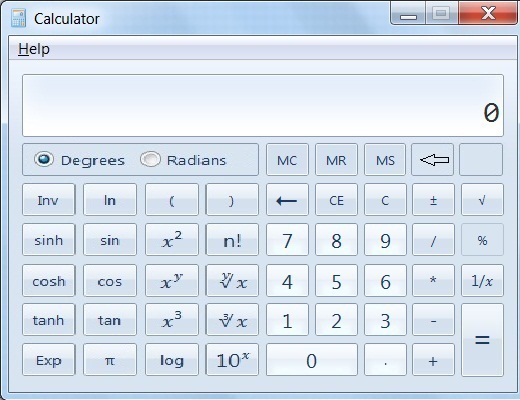
Điểm cần lưu ý là trong phần cứng (hardware) của đa số điện thoại cầm tay không có bộ xử lý tính dấu phẩy động (floating-point processor). Điều này làm việc phát triển phần mềm cho chúng trở nên phức tạp và lý thú hơn. Cũng nên biết rằng rất nhiều ứng dụng phần mềm đòi hỏi phép tính dấu phẩy động (bề ngoài tưởng là không cần thiết), ví dụ như tính toán các phần cong (curves) cho các phong chữ DGOPSWY… Để xử lý số thực SV sẽ tìm hiểu và áp dụng **fixed-point** vào tất cả phép tính toán.

Phần mềm máy tính bao giờ cũng được cài đặt sẵn trong bất cứ điện thoại cầm tay nhưng chỉ dừng lại ở mức độ cơ bản như cộng trừ nhân chia, trong khi đó mức nhu cầu sử dụng của giới làm việc “trí thức” cao hơn, ở mức độ của máy tính khoa học.

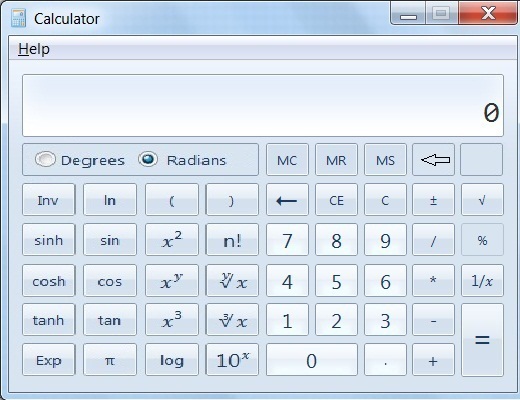
Đề tài “*Xây Dựng Phần Mềm Ứng Dụng Máy Tính Khoa Học cho Điện Thoại Cầm Tay Thông Minh chạy trên hệ điều hành Android”* nhằm nâng cao kỹ năng lập trình, cũng như phân tích, thiết kế và xây dựng một ứng dụng thực tiển cho sinh viên từ những yêu cầu được phân tích như trên. Xây dựng được một phần mềm ứng dụng như một máy tính khoa học thực thụ sẽ không đơn giản. Cho nên mức độ yêu cầu của đề tài chỉ bao gồm những phép toán và hàm đơn giản. Đồng thời ứng dụng được giới hạn chỉ cho máy điện thoại cầm tay thông minh do màn hình (lớn, cảm ứng, …) của chúng sẽ cho phép thiết kế và xây dựng một giao diện phức tạp đa năng, và bộ xử lý của chúng đủ mạnh để tính toán bài toán phức tạp (phát sinh từ việc không có bộ xử lý tính dấu phẩy động) nhằm đáp ứng được yêu cầu của đề tài.

## YÊU CẦU ĐỒ ÁN VÀ THANG ĐIỂM

1. Xây dựng được giao diện (graphics interface) cho ứng dụng trên nền tảng Android giống như các hình sau **(10%)**:
   1. Để tính các hàm x2, x3, , 1/x, n!, ln (log cơ số e), log (log cơ số 10), 10x, sin, cos, tan, xy, , sinh, cosh, tanh và góc lượng giác có đơn vị là độ (0):

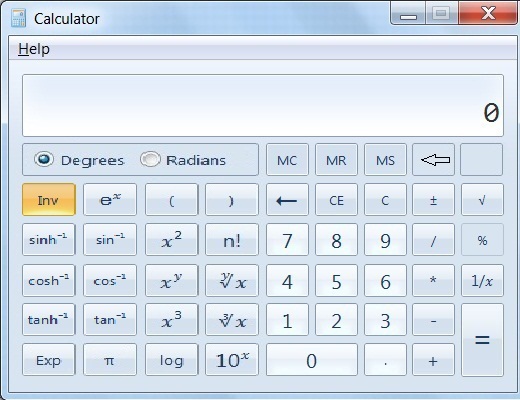


* 1. Để tính các hàm x2, x3, , 1/x, n!, ln (log cơ số e), log (log cơ số 10), 10x, sin, cos, tan, xy, , sinh, cosh, tanh và góc lượng giác có đơn vị là radians (rad):



inv

* 1. Bấm nút để tính các hàm ex, sin-1 (arcsin), cos-1 (arcos), tan-1 (arctan), sinh-1 (arcsinh), cosh-1 (arcosh), tanh-1 (arctanh) và góc lượng giác có đơn vị và độ (0):



**Lưu ý**: SV phải giao diện như hình trên để tránh trường hợp SV lấy phần mềm có sẵn trên mạng và xem là sản phẩm của mình làm ra. **Nếu SV không thực hiện đúng yêu cầu này SV sẽ bị điểm KHÔNG cho đồ án.**

1. Nhập và xử lý biểu thức **(10%)**:
   1. Cho phép nhập biểu thức dùng dạng mẩu (inputting expression using standard format) như của một máy tính khoa học thực sự. Ví dụ có thể nhập biểu thức:

2\*(3.6 + log(5 + 3÷4.1) – sin(29.7)) – e2.7

sin

* 1. Nhập hàm: tự động thêm dấu ngoặc mở khi nhập hàm. Ví dụ nhấn hàm sẽ thể hiện sin( trong biểu thức
  2. Cho phép bỏ dấu nhân (\*) trong các trường hợp sau:
     1. Trước dấu ngoặc mở ‘(‘. Ví dụ: 2\*(3.6 + 5.6) → 2(3.6 + 5.6)
     2. Trước hàm có dấu ngoặc mở: 2.0\*sin(30) → 2.0sin(30)
     3. Trước một biến số, hằng số: 2A, 2y, 2π, …

1. backcursorforwardcursorSửa chữa biểu thức: có thể di chuyển con trỏ (cursor) để xóa/thêm trên biểu thức. Cụ thể dùng các nút như sau **(5%)**:
   1. CEbackspaceC và để di chuyển lui và tới cho con trỏ (move backward and forward the cursor)
   2. , , và để xóa 1 chữ số, 1 con số, và cả biểu thức.
2. =Gọi và dùng lại nội dung của (ít nhất năm) biểu thức tính toán cũ **(5%)**
   1. Tự động lưu biểu thức vào bộ nhớ sau bấm
   2. MRMS để lưu biểu thức hiện có trên màn hình
   3. MC để thể hiện biểu thức củ
   4. để xóa hết biểu thức trong bộ nhớ
3. Các phép tính cơ bản cộng +, trừ -, nhân \*, chia ÷ **(10%)**
4. Các phép tính với các hàm: x2, x3, , 1/x, n!, ln (log cơ số e), ex, log (log cơ số 10), 10x, sin, cos, tan, xy, , sinh, cosh, tanh **(tối đa 50%: mổi hàm được tính 5%)**
5. Các phép tính với các hàm: sin-1, cos-1, tan-1, sinh-1, cosh-1, tanh-1 **(tối đa 15%: mổi hàm được tính 5%)**
6. Có thể dùng degree hay radians cho các hàm lượng giác (trigonometry) **(5%)**
7. Các phép tính dựa trên nền sáu mươi (sexagesminal) của độ, phút, giây. Ví dụ: 205’35” **(5%)**
8. Biểu thị kết quả: **(5%)**
   1. Dạng số thập phân (decimal number), ví dụ 100÷7 = 14.2857142857142857142. Có thể xác định số chữ số thập phân (decimal place) của kết quả, ví dụ 100÷7 = 14.286 với 3 chữ số thập phân
   2. Dạng số khoa học (scientific number) ví dụ 100÷7 = 1.42857142857142857142E+01. Đồng thời cũng có thể xác định số chữ số có nghĩa (significant digits) của kết quả, ví dụ 100÷7 = 1.4286E+01 với 5 số chữ số có nghĩa
9. Hướng dẫn sử dụng **(5%)**

**Lưu ý:**

1. SV Không được sử dụng thư viện math.h có sẵn cho các phép tính của số thực của các yêu cầu 5, 6, 7, 8, 9. SV **phải dùng fixed-point** cho các phép tính này. Nếu không làm đúng yêu cầu này thì điểm của các yêu cầu này sẽ bị điểm **không**.
2. Tổng số điểm yêu cầu đạt được là 100%

# NỘI DUNG CÁO BÁO

* + 1. DỰ KIẾN BAN ĐẦU:

|  |  |
| --- | --- |
| **Tuần** | **Dự kiến** |
| 1,2,3 | Nghiên cứu ngôn ngữ Java + phần mềm Eclipse |
| 4 | Cài đặt Eclipse + viết các module đơn giản |
| 5 |
| 6 | Operator : +,-, \*, /,%,!,^ |
| 7 |
| 8 | Ký pháp Ba Lan |
| 9 | Giải phương trình, Log, ln, e, đổi cơ số |
| 10 | Fixed-point |
| 11 |
| 12 | Các hàm: Luong giac, Ln, e, can |
| 13 |
| 14 | Tích hợp fixed-point vào các module |
| 15 | Xong báo cáo và nộp đồ án |

**Bảng 1: Dự kiến ban đầu**

Nhóm chia thành 3 nhóm nhỏ:

Nhóm Cấu trúc: Luân, Thanh

Nhóm Phân tích: Thanh, Luân

Nhóm Nghiên cứu, testing: Hoàng, Tân

* + 1. QUÁ TRÌNH PHÂN CÔNG THỰC HIỆN:

|  |  |  |  |
| --- | --- | --- | --- |
| **Thanh** | **Luân** | **Hoàng** | **Tân** |
| Thiết kế giao diện và các chức năng hiển thị của giao diện. | Thưc hiện code fixpoint, balan và các hàm cơ bản. | Nghiên cứu và đưa ra giải thuật cho các phép toán. | Nghiên cứu và thực hiện báo cáo. |
| Thực hiện code các hàm lượng giác theo chuỗi Talor | Thực hiện code các hàm căn thức, lũy thừa, … | Thực hiện code kiểm tra biểu thức. | Code các nút chức năng MR, MS, MC |

**Bảng 2: Quá trình phân công công việc**

* + 1. CHỨC NĂNG PHẦN MỀM:

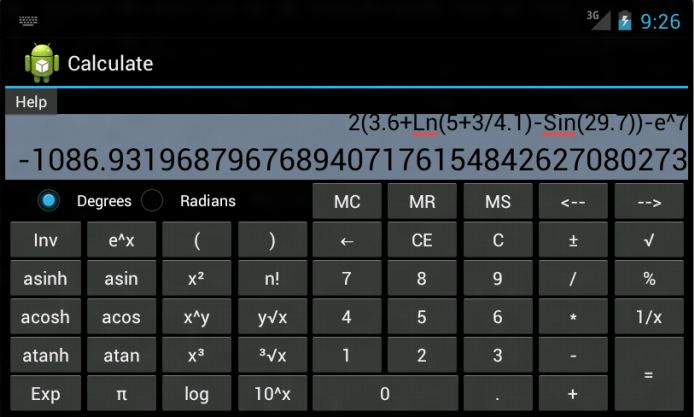
Sau khi hoàn tất, phần mềm có khả năng thực hiện những chức năng sau:

* + - 1. Nhập biểu thức:

Cho phép nhập biểu thức dùng dạng mẩu (inputting expression using standard format) như của một máy tính khoa học thực sự.

Ví dụ có thể nhập biểu thức:

2(3.6 + ln(5 + 3÷4.1) – sin(29.7)) – e7

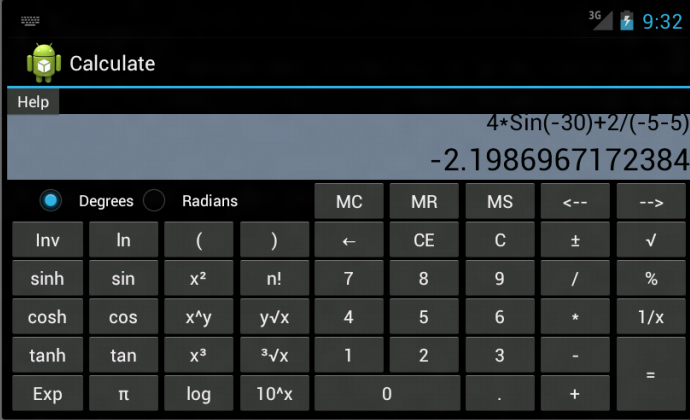


**Hình 1: Nhập biểu thức**

* + - 1. Nhập hàm:   
         * Tự động thêm dấu ngoặc mở khi nhập hàm. Ví dụ nhấn hàm sẽ thể hiện sin( trong biểu thức.

sin

* + - * + Cho phép bỏ dấu nhân (\*) trong các trường hợp sau:
* Trước dấu ngoặc mở ‘(‘. Ví dụ: 2\*(3.6 + 5.6) → 2(3.6 + 5.6)
* Trước hàm có dấu ngoặc mở: 2.0\*sin(30) → 2.0sin(30)
  + - * + Cho phép nhập - ở đầu biểu thức (vd:, -2 ), ở sau dầu ‘(‘. (vd: 4+(-2\*3)



**Hình 2: Nhập hàm**

* + - * + Tự động kiểm tra và báo lỗi một số biểu thức khi nhập. ( Chưa thưc hiện kiểm tra và xuất ra lỗi)
      1. Các phép tính cơ bản cộng +, trừ -, nhân \*, chia /, chia lấy dư %,lủy thừa.
      2. Các phép tính với các hàm: n!, ln (log cơ số e),log, ex( với x >0), sin, cos, tan,và các hàm lượng giác khác.
      3. Các phép tính với các hàm: x^y ( y là nguyên) và Căn bậc n của 1 số.
      4. Xây dựng giao diện tương tự như đồ án yêu cầu
    1. HƯỚNG DẪN SỮ DỤNG

### Tổng quát

Đây là chương trình máy tính được lập trình chạy trên điện thoại hệ điều hành android với nền tảng android trong lúc lập trình là 2.3.3

Khi chạy chương trình hệ thống sẽ hiện ra 1 bảng các nút nhấn cần thiết để tính toán biểu thức và các nút ẩn sẽ được hiển thị khi thực hiện các chức năng khác.

### Chi tiết

Sau đây là một số hướng dẫn cách sử dụng phần mềm máy tính:



Hình 3: Giao diện máy tính

### Các nút chuyển đổi

### Inv

Là nút chuyển đổi giá trị của một số nút. Sau khi nhấn nút Inv, giá trị các nút sau đây sẽ thây đổi:

|  |  |
| --- | --- |
| Ban đầu | Sau đó |
| ln | e^ |
| sin | asin |
| cos | acos |
| tan | atan |
| sinh | asinh |
| cosh | acosh |
| tanh | atanh |

### Chuyển đổi góc

Các nút này được áp dụng cho các hàm lượng giác gồm các nút: Deg, Rad.

Mặc định máy tính luôn ở nút Deg.

Trước tiên, người dùng chọn nút muốn chuyển đổi sang rùi nhập giá trị tính của các hàm lượng giác, cuối cùng là nhấn dấu ‘=’ để xem kết quả.

Ví dụ:

Deg: Sind(45) = 0.7.710807

Rad: Sinr(90) = 1.0000…

* + - 1. Nhập biểu thức:
* Được phép nhập 1 biểu thức dùng dang mẫu.
* Đối với các phép toán 2 ngôi (\* / % can ^ ) cần nhập theo dạng a^b.
* Đối với phép toán - có thể nhập -a.
* Chỉ đc nhập 2 phép toán kế liền nhau khi phép toán sau là phép + hoặc –

( vd: 3\*-2 ).

* Đước phép bỏ dấu \* trong các trường hợp sau:
* Trước dấu ngoặc 3( -> 3\*(
* Trước 1 hàm 3sin( -> 3\*sin(

### Phép tính với hàm

### Hàm lượng giác

Gồm có các hàm sau**: sin, cos, tan, sinh, cosh, tanh, asin, acos, atan, asinh, acosh, atanh**. Được tính bằng cách nhấn nút hàm cần tính rùi nhập giá trị vào và nhấn dấu ‘=’ để xem kết quả cuối cùng.

Trong đó:

* Các hàm sin, cos, tan được tính bằng đơn vị độ.
* Hàm asin, acos, atan là các hàm lượng giác ngược được tính bằng arc của sin, cos, tan có giá trị nằm trong khoảng -1 🡪1.
* Hàm sinh, cosh có giá trị nằm trong khoảng -1 🡪1, tanh có giá trị nằm trong khoảng 0 🡪1

Ví dụ: sin(90) = 1.000000000031090

cos(0) = 1

asin(0.5) = 30.456859643383  
**Chú ý**: cần nhấn dấu ngoặc đóng ‘)’ cuối biểu thức để tính, không có chường trình vẽ bị thoát.

### Hàm căn bậc 2

Gồm các . Người dùng bấm các nút căn trước rồi nhập giá trị cần tính vào, cuối cùng là nhấn dấu ‘=’ để xem kết quả.

Ví dụ: = 4.0000000000

### Hàm căn bậc n

Chưa thực hiện được

### Hàm mũ

Gồm các hàm **x2, x3, xn, 10n, ex.** Người dùng cần nhấn giá trị cần tính trước rồi mới nhấn nút hàm mũ, nếu không đúng máy tính sẽ báo lỗi.

Ví dụ: 103 = 1000.

**Chú ý:** chưa thực hiện được với số thực

### Hàm logarit

Gồm có các hàm **Ln, Llog**. Người dùng nhấn nút hàm tính logarit rồi nhập giá trị tính vào, sau đó nhấn nút dấu ‘=’ để xem kết quả.

* Ln: trả về logarit tự nhiên được tính dựa vào hằng số e.
* Log: trả về kết quả logarit cơ số 10 của một số.

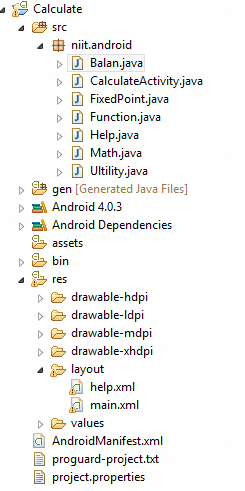
Ví dụ: log(100) = 1.999999050043617

### Một số nút chức năng khác

Gồm các nút: **MC, MR, MS, 🡨, -->, <-- CE, C** được thể hiện màu đỏ có tác dụng sau.

* C: xóa toàn bộ chuỗi biểu thức được nhập vào.
* CE: xóa chuỗi số vừa được nhập vào, nếu gặp các phép tính hay dấu ‘(‘, ‘)’ thì sẽ không thể thực hiện được.
* 🡨: xóa 1 ký tự phía trước con trỏ bất kể là số hay dấu phép tính.
* MS: lưu kết quả vào bộ nhớ
* MR: truy cập vào bộ nhớ và gọi kết quả lưu của M+.
* MC:xóa giá trị lưu vào bộ nhớ của M+.
* <-- : di chuyển lùi trên ô nhập biểu thức.
* -->: di chuyển tới trên ô nhập biễu thức.
  + 1. MÔ TẢ PHẦN MỀM:

*Chương trình gồm 7 file java như trong hình: Balan.java, CalculateActivity.java, FixedPoint.java, Function.java, Help.java, Math.java, Ultility.java và 2 file giao diện là main.xml và help.xml*



Hình 4: Cấu trúc chương trình Calculator

1. Ý TƯỞNG CODE, ÁP DỤNG CÁC CÔNG THỨC VÀ THUẬT TOÁN  
   * + 1. CalculateActivity.java :

**public** **class** calculator **extends** Activity **implements** OnClickListener{}

Đây là class sử dụng để bắt sự kiện các nút Button,RadioButton,CheckBox ,hiển thị biểu thức lên TextBox1 và hiển thị kết quả lên text box 2,đồng thời class calculator còn có nhiệm vụ quan trọng là xữ lý sự kiện cho từng module riêng biệt thông qua việc gọi hàm từ các class khác.

* 1. Khai báo và bắt sự kiện :

Đoạn code sau trên dùng đễ xác định các địa chĩ button với các kí tự tương ứng:

…

**final** Button help = (Button) findViewById(R.id.*B\_Help*);

**final** EditText stack = (EditText) findViewById(R.id.*stack*

**final** TextView out = (TextView) findViewById(R.id.*out*);

Button khong = (Button) findViewById(R.id.*B\_0*);

**final** Button mot = (Button) findViewById(R.id.*B\_1*);

…

* 1. Xử lý sự kiện :

…

//--------------------- xữ lý button pi----------------------------------

pi.setOnClickListener(**new** View.OnClickListener() {

**public** **void** onClick(View v) {

stack.getText().insert(stack.getSelectionStart(), "pi"); }

});

}

});

…

// hàm sử lý dấu bằng củng với các xử lý của nó

bang.setOnClickListener(**new** View.OnClickListener() {

**public** **void** onClick(View v) {

out.setText("0");

Balan b = **new** Balan();

String test = b.HamTinhToan(b.RPN(stack.getText().toString **boolean** j = **false**;

**int** d = 0;

**try** {

**for** (**int** i = 0; i < test.length(); i++) {

**if** (test.charAt(i) != '.' && test.charAt(0) != '-' {

**if** (!Character.*isDigit*(test.charAt(i {

j = **true**

**break**;

}

}

**else**

{

**if** (test.charAt(i) == '.')

{

d++;

}

}

}

**if** (j == **true** || d > 1)

{

stack.setText("Math Error);

} **else** **if** (j == **false**) {

out.setText(test);

}

} **catch** (Exception e) {

stack.setText("Math Error or Invalid Expression”);

}

}

});

…

Khi nhấn nút = hàm sẽ kiểm đưa biểu thức vào balan để tính toán sau đó kiểm tra kết quả một lầ nữa và xuất ra kết quả lại ô xuất của máy tính.

* 1. Sử lý sự kiên button Inv

Bằng cách thay đổi hiễn thị chữ trên các button có thay đổi. Phương pháp này sẽ tối ưu và nhanh hơn phương pháp sử dụng 1 giao diện mới thay thế.

Nhưng khi bấm vào thì hiễn thị trên khung hiện thị vẫn là hiện thị của button cũ. Ta sử lý bằng cách gắn vào 1 cờ TamChange.

Sau đó sẽ sử dụng cờ TamChange để sử lý trên các button có thay đổi.

**VD:** Xử lý Button Sin

sin.setOnClickListener(**new** View.OnClickListener() {

**public** **void** onClick(View v) {

**if** (TamChange == 0) {

stack.getText().insert(stack.getSelectionStart(), "Sin(");

} **else** {

stack.getText().insert(stack.getSelectionStart(), "Asin(");

}

}

});

* 1. Sử lý sự kiên button MS, MR, MC

Sử dụng stack để lưu chuỗi cần lưu và gọi lại ô sử lý khi cần thiết. Nếu không cần cùng đến có thể xóa khỏi stack

* + - 1. FixedPoint.java :
* Class Fixed Point được dùng để xử lý dữ liệu đầu vào là một chuỗi ký tự số với độ dài có thể rất lớn trong đó có chứa dấu chấm thập phân.
* Fixed Point được định nghĩa là một chuỗi số thực với thuộc tính bao gồm một danh sách liên kết kiểu ký tự, một biến kiểu integer dùng để lưu vị trí dấu chấm thập phân, một biến kiểu boolean dùng để lưu giá trị âm hoặc dương (mặc định false là dấu “+”, true là dấu “-”).
* Chuỗi số thực sau khi được định nghĩa thì vị trí trước dấu chấm thập phân của chuỗi được xem là hàng đơn vị, tiếp theo là hàng chục, hàng trăm…của phần chuỗi số nguyên, vị trí sau dấu chấm được xem là phần chuỗi số thập phân.
* Phương thức khởi tạo Fixed Point gồm có hai loại :
* Phương thức khởi tạo với dữ liệu đầu vào là chuỗi kí tự số.

Phương pháp xử lý:

* Chuyển dữ liệu đầu vào thành một mảng ký tự số, xét dấu mặc định của chuỗi số thực là dấu “+”, kiểm tra từ đầu chuỗi đến cuối chuỗi của dữ liệu đầu vào, đưa từng ký tự vào danh sách liên kết (nếu gặp dấu “-” ngay từ đầu chuỗi ký tự thì xét dấu true).
* Sau đó chuyển tất cả các ký tự số còn lại vào danh sách liên kết (kể cả dấu “.”). Nếu trong chuỗi không có ký tự là dấu “.” thì sẽ thêm dấu “.” và số 0 vào danh sách liên kết.
* Nếu phần tử cuối của danh sách liên kết là dấu “.” thì sẽ thêm số 0 vào cuối danh sách liên kết.
* Xét lại danh sách liên kết và loại bỏ phần tử có giá trị là kí tự ‘0’ dư ở đầu và cuối danh sách liên kết.
* Xét biến dùng để lưu vị trí dấu chấm thập phân = vị trí của dấu “.” trong danh sách liên kết.

Ví dụ: người dủng nhập vào 000123736358 -> fixed point: 123736358.0 (vị trí dấu chấm = 9, giá trị âm dương = false(dương))

* Phương thức khởi tạo không có dữ liệu đầu vào
* Phương thức này sẽ tạo ra một danh sách liên kết rỗng không có phần tử kí tự số. Các biến lưu vị trí dấu chấm, dấu âm dương (+,-) sẽ được thiết lập mặc định.
* Mục đích sử dụng của phương thức này là tạo ra một chuỗi số thực tạm để chúng ta có thể gán một chuỗi số thực nào đó vào chuỗi tạm này trong quá trình xử lý ( có thể xem như là một biến nhớ ).
  + - 1. Balan.java:

Hiện nay, việc tính giá trị một biểu thức toán học là điều quá đỗi bình thường. Tuy nhiên, trong nhiều ứng dụng (như chương trình vẽ đồ thị hàm số chẳng hạn, trong đó chương trình cho phép người dùng nhập vào hàm số), ta cần phải tính giá trị của một biểu thức được nhập vào từ bàn phím dưới dạng một chuỗi. Với các biểu thức toán học đơn giản (như a+b) thì bạn có thể tự làm bằng các phương pháp tách chuỗi “thủ công”. Nhưng để “giải quyết” các biểu thức có dấu ngoặc, ví dụ như (a+b)\*c + (d+e)\*f ,  thì các phương pháp tách chuỗi đơn giản đều không khả thi. Trong tình huống này, ta phải dùng đến Ký Pháp Nghịch Đảo Ba Lan (Reserve Polish Notation – RPN).

**Ví dụ :**

Cho biểu thức trung tố :

5 + ((1 + 2) \* 4) + 3

được biểu diễn lại dưới dạng hậu tố là (ta sẽ bàn về thuật toán chuyển đổi từ trung tố sang hậu tố sau) :

5 1 2 + 4 \* + 3 +

Quá trình tính toán sẽ diễn ra theo như bảng dưới đây :

|  |  |  |
| --- | --- | --- |
| **Ký tự** | **Thao tác** | **Trạng thái stack** |
| 5 | Push 5 | 5 |
| 1 | Push 1 | 5, 1 |
| 2 | Push 2 | 5, 1, 2 |
| + | Tính 1 + 2  Push 3 | 5, 3 |
| 4 | Push 4 | 5, 3, 4 |
| \* | Tính 3 \* 4  Push 12 | 5, 12 |
| + | Tính 12 + 5  Push 17 | 17 |
| 3 | Push 3 | 17, 3 |
| + | Tính 17 + 3  Push 20 | 20 |

**Bảng 3: Ví dụ tính toán trên chuỗi hậu tố**

* **Chuyển đổi từ trung tố sang hậu tố**

Thuật toán này cũng dựa theo cơ chế ngăn xếp. Ý tưởng chung của thuật toán cũng là duyệt biểu thức từ trái sang phải:

- Nếu gặp một toán hạng (con số hoặc biến) thì ghi nó vào chuỗi kết quả (chuỗi kết quả là biểu thức trung tố).

- Nếu gặp dấu mở ngoặc, đưa nó vào stack.

- Nếu gặp một toán tử (gọi là o1 ), thực hiện hai bước sau:

o Khi nào còn có một toán tử o2 ở đỉnh ngăn xếp và độ ưu tiên của o1 nhỏ hơn hay bằng độ ưu tiên của o2 thì lấy o2 ra khỏi ngăn xếp và ghi vào kết quả.

o Push o1 vào ngăn xếp

- Nếu gặp dấu đóng ngoặc thì cứ lấy các toán tử trong ngăn xếp ra và ghi vào kết quả cho đến khi lấy được dấu mở ngoặc ra khỏi ngăn xếp.

- Khi đã duyệt hết biểu thức trung tố, lần lượt lấy tất cả toán hạng (nếu có) từ ngăn xếp ra và ghi vào chuỗi kết quả.

Để dễ hiểu, chúng ta hãy quan sát quá trình thực thi của thuật toán qua một ví dụ cụ thể sau:

Biểu thức cần chuyển đổi: 3+4\*2/(1-5)

|  |  |  |  |
| --- | --- | --- | --- |
| **Ký tự** | **Thao tác** | **Stack** | **Chuỗi hậu tố** |
| 3 | Ghi 3 vào k.quả |  | 3 |
| + | Push + | + |  |
| 4 | Ghi 4 vào k.quả |  | 3 4 |
| \* | Push \* | + \* |  |
| 2 | Ghi 2 vào kquả |  | 3 4 2 |
| / | Lấy \* ra khỏi stack, ghi vào k.quả, push / | +  / | 3 4 2 \* |
| ( | Push ( | + / ( | 3 4 2 \* |
| 1 | Ghi 1 vào k.quả | + / ( | 3 4 2 \* 1 |
| - | Push - | + / ( – | 3 4 2 \* 1 |
| 5 | Ghi 5 vào k.quả | + / ( – | 3 4 2 \* 1 5 |
| ) | Pop cho đến khi lấy được (, ghi các toán tử pop được ra k.quả | + / | 3 4 2 \* 1 5 - |
| 2 | Ghi 2 ra k.quả | + / | 3 4 2 \* 1 5 – 2 |
|  | Pop tất cả các toán tử ra khỏi ngăn xếp và ghi vào kết quả |  | 3 4 2 \* 1 5 – 2  / + |

**Bảng 4: Mô tả thuật toán ký pháp BaLan**

* + - 1. Math.java:

Chứa 2 các hàm phép toán cơ bản +, -, \* / và các các hàm hỗ trợ thực hiện các phép toán đó.

Hàm Cộng ( xử lý phép cộng)

* Dữ liệu đầu vào là hai chuỗi ký tự số thực.

**Ví dụ:** 656.70 + 05.124

* Xử lý hai chuỗi số thực nhằm đưa chúng về dạng chuẩn của số thực như sau:

656.70 🡪 656.7

05.124 🡪 5.124

* Thực hiện cân bằng chiều dài của hai chuỗi số thực bằng cách thêm kí tự ‘0’ vào đầu( hoặc cuối) hai chuỗi kí tự số thực trong hai trường hợp sau :
* Ta sẽ thêm kí tự ‘0’ vào đầu hai chuỗi số thực nếu phần nguyên của hai chuỗi số thực không bằng nhau.
* Thêm kí tự ‘0’ vào cuối hai chuỗi số thực nếu phần thập phân của hai chuỗi số thực không bằng nhau.

**Ví dụ :** 656.7 656.700

+5.124 + 005.124

* Thực hiện cộng giá trị của từng phần tử ở hai chuỗi số thực lại với nhau, bắt đầu từ phần tử nằm ở vị trí cuối cùng của hai chuỗi số thực ,lần lượt cộng đồng thời gán giá trị nhớ vào một biến nhớ như phép cộng tay và lưu giá trị tính được vào kết quả, nếu gặp dấu “.” thì thêm dấu “.” vào kết quả.

**Ví dụ:**

|  |  |  |  |
| --- | --- | --- | --- |
| Số 1 | Số 2 | Số nhớ | Kết quả |
| 0 | 4 | 0 | 4 |
| 0 | 2 | 0 | 2 |
| 7 | 1 | 0 | 8 |
| . | . | 0 | . |
| 6 | 5 | 1 | 1 |
| 5 | 0 | 0 | 6 |
| 6 | 0 | 0 | 6 |

**Bảng 5: Phép Cộng**

Hàm Trừ ( xử lý phép trừ)

* Dữ liệu đầu vào là 2 chuỗi kí tự số thực.

**Ví dụ:** 656.70 - 05.124

* Xử lý 2 chuỗi số thực nhằm đưa chúng về dạng chuẩn của số thực và tạm gọi chúng là số trừ và số bị trừ :

656.70 🡪 656.7 (Số trừ)

05.124 🡪 5.124 (Số bị trừ)

* Thực hiện cân bằng chiều dài của hai chuỗi số thực bằng cách thêm kí tự ‘0’ vào đầu( hoặc cuối) hai chuỗi kí tự số thực trong hai trường hợp sau :
* Ta sẽ thêm kí tự ‘0’ vào đầu hai chuỗi số thực nếu phần nguyên của hai chuỗi số thực không bằng nhau.
* Thêm kí tự ‘0’ vào cuối hai chuỗi số thực nếu phần thập phân của hai chuỗi số thực không bằng nhau.

**Ví dụ :** 656.7 656.700

-5.124 - 005.124

* Thực hiện so sánh hai chuỗi số thực, nếu số nào lớn hơn thì lấy số đó trừ số nhỏ hơn và đồng thời xét dấu của kết quả là âm.
* Thực hiện trừ từng phần tử giữa số trừ và số bị trừ, bắt đầu từ phần tử ở vị trí cuối cùng của hai chuỗi.
* Thực hiện trừ cho đến hết phần tử ở vị trí đầu tiên như phép trừ tay, đồng thời gán giá trị nhớ vào một biến nhớ và lưu giá trị thu được vào kết quả, sau đó sửa lại kết quả nếu dư 0 ở đầu và cuối của chuỗi kết quả, xét vị trí dấu chấm thập phân và lưu dấu chấm vào kết quả.
* **Ví dụ:**

|  |  |  |  |
| --- | --- | --- | --- |
| Số trừ | Số bị trừ | Số nhớ | Kết quả |
| 0 | 4 | 1 | 6 |
| 0 | 2 | 1 | 7 |
| 7 | 1 | 0 | 5 |
| . | . | 0 | . |
| 6 | 5 | 0 | 1 |
| 5 | 0 | 0 | 5 |
| 6 | 0 | 0 | 6 |

**Bảng 6: Phép Trừ**

Hàm Nhân ( xử lý phép nhân)

* Dữ liệu đầu vào là hai chuỗi kí tự số thực.

**Ví dụ:** 656.70 x 05.124

* Xử lý 2 chuỗi số thực nhằm đưa chúng về dạng chuẩn của số thực và tạm gọi chúng là số 1 và số 2.

656.70 🡪 656.7 (Số 1)

05.124 🡪 5.124 (Số 2)

* Dùng một biến kiểu integer và lưu tổng chiều dài phần thập phân của số 1, 2 .

**Ví dụ:** 656.70 x 05.124 🡪 dot = 4

* Thực hiện cân bằng chiều dài của cả hai số bằng cách thêm kí tự ‘0’ vào đầu hoặc cuối chuỗi kí tự ở cả 2 số.

**Ví dụ :** 656.7 656.700

x 5.124 x 005.124

* Xóa dấu chấm thập phân ở cả hai chuỗi số thực để tiến hành phép nhân.

656.700 656700

x 005.124 x 005124

* Lần lượt nhân tất cả các kí tự số ở số 1 với từng kí tự số ở số 2 (bắt đầu từ phần tử ở vị trí cuối của số 2) và lưu giá trị thu được vào một số tạm .
* Sau đó thêm kí tự ‘0’ vào sau vị trí cuối cùng của chuỗi số tạm nhằm cân bằng chiều dài cho phép cộng sau này.
* Tiếp tục nhân số 1 cho các phần tử tiếp theo của số 2 cho đến khi hết chuỗi kí tự số của số 2, đồng thời cộng dồn kết quả của số tạm.

656700

x 005124

2626800

+ 13134000

65670000

3283500000

3364930800

* Xét vị trí dấu chấm thập phân và dấu âm dương của kết quả.

dot = 4 🡪 3364.930800

* Kết quả trả về cuối cùng là chuỗi số thực mang ý nghĩa là tích giá trị của số 1 và số 2.

Hàm Chia (xử lý phép chia)

* Dữ liệu đầu vào là hai chuỗi kí tự số thực.
* Xử lý 2 chuỗi số thực nhằm đưa chúng về dạng chuẩn của số thực và tạm gọi chúng là số chia và số bị chia.
* Định nghĩa một biến kiểu integer dùng để giới hạn độ dài của kết quả nếu kết quả là một chuỗi số thực vô hạn.
* Thực hiện cân bằng phần thập phân của số chia và số bị chia, sau đó bỏ dấu chấm thập phân ở cả hai số.

**Ví dụ:** 345.6 / 78.679 -> 345600 / 78679

* Kiểm tra xem một trong hai số chia và số bị chia có “mang” giá trị bằng “0” hay không (nghĩa là chuỗi số thực sau khi xử lý tương đương với chuỗi “0.0”).
* Nếu số chia bằng “0” thì sẽ trả về kết quả là “0.0”. Nếu số bị chia bằng “0” sẽ trả về kết quả là “Not a number”.
* Kiểm tra xem nếu số chia lớn hơn số chia thì ta sẽ ghi giá trị “0” vào chuỗi số kết quả, sau đó thêm “0” vào số bị chia cho đến khi số chia lớn hơn số chia thì dừng.
* Sau đó thêm dấu chấm thập phân sau kí tự “0” đầu tiên trong chuỗi kết quả.
* Sau đó tiếp tục phép chia với điều kiện lúc này là số chia đã lớn hơn số bị chia, đồng thời dùng 1 biến kiểu boolean để đánh dấu là phép chia này đã có dấu chấm thập phân ( True: đã có dấu chấm, False : chưa có dấu chấm ).

**Ví dụ :** 3 / 87

|  |  |  |
| --- | --- | --- |
| **Bước** | **Số bị chia** | **Kết quả** |
| 1 | Kiểm tra (3 < 87) 🡪 30 | 0 |
| 2 | Kiểm tra (30 < 87) 🡪 300 | 0.0 |
| 3 | Kiểm tra (300 > 87) 🡪dừng | 0.0 🡪tiếp tục phép chia bên dưới với số chia > số bị chia |

Bảng 7: Mô tả phép chia

* Ngược lại nếu số chia bằng số chia, ta sẽ trả về kết quả là “1.0”
* Nếu số bị chia lớn hơn số chia :
* Ta dùng một chuỗi số tạm với tên gọi là “số chia tạm” để lấy n kí tự số trong số bị chia với n bằng chiều dài chuỗi kí tự của số chia, một chuỗi số thực với tên gọi là “dư” để lấy n kí tự số còn lại trong chuỗi kí tự của số chia.
* Thực hiện phép chia bằng cách lặp lại n lần những bước sau đây:
* Kiểm tra xem chiều dài chuỗi số kết quả có bằng giá trị của biến giới hạn đã định nghĩa ở bước trên (giới hạn chiều dài chuỗi kí tự số của chuỗi số kết quả ) hay không, hoặc số chia tạm có bằng “0” hay không (số bị chia tạm bằng “0” có nghĩa là phép chia hết). Nếu không rơi vào một trong hai trường hợp trên ta thực hiện các bước tiếp theo.
* Nếu số chia tạm có giá trị nhỏ hơn số chia, ta sẽ lấy một kí tự số đầu của số “dư” thêm vào sau vị trí cuối cùng của chuỗi số chia tạm. Trường hợp số “dư” đã rỗng ( có thể do chiều dài của chuỗi số bị chia bằng số chia hoặc sau một số n lần lặp thực hiện phép chia đã lấy hết phần tử trong số “dư”) ta sẽ thêm kí tự ‘0’ vào cuối chuỗi số bị chia tạm, và thêm dấu chấm thập phân vào chuỗi kết quả, đồng thời kiểm tra xem biến kiểu Boolean ở trên đã ghi nhận lại dấu chấm thập phân hay chưa, nếu chưa ta sẽ thêm dấu chấm thập phân vào chuỗi kết quả và ghi nhận giá trị True vào biến này.
* Sau khi số chia tạm đã lớn hơn số chia, ta sẽ thực hiện phép trừ số chia tạm cho số chia, ta sẽ dùng 1 biến “i” kiểu int để ghi nhận lại số lần trừ cho đến khi số bị chia tạm nhỏ hơn số bị chia.
* Ghi giá trị của biến “i” ở bước trên vào kết quả.
* Lặp lại bước 1.
* Sau khi thực hiện phép chia, nếu số “dư” vẫn chưa rỗng và chiều dài chuỗi kết quả vẫn chưa vượt quá giới hạn cho phép, ta sẽ đưa tất cả các kí tự số còn lại ở số “dư” thêm vào sau vị trí cuối cùng của chuỗi kết quả.
* Nếu sau khi thực hiện phép chia mà vẫn chưa có dấu chấm thập phân, ta sẽ thêm dấu chấm thập phân vào cuối chuỗi kết quả.

**Ví dụ :** 80 / 25

|  |  |
| --- | --- |
| Số chia | Số bị chia |
| 80 | 25 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Số chia tạm | Số dư | i | checkDot | Kết quả |
| 80 | null | 3 | false | 3 |
| 50 | null | 2 | true | 3.2 |

* + - 1. Function.java :

Class dùng đễ tính các phép toán lượng giác và hàm lập phương (x3), hàm bình phương(x2), hàm ngũ bậc n, hàm giai thừa (x!), hàm căn bậc 2 , hàm căn bậc n, hàm log, hàm ln, đổi từ radian sang độ, hàm đổi từ độ sang radian.

**Các hàm đổi góc trước khi sử dụng tuật toán Taylor để tính toán :**

* + Công thức chuyển từ độ sang radian

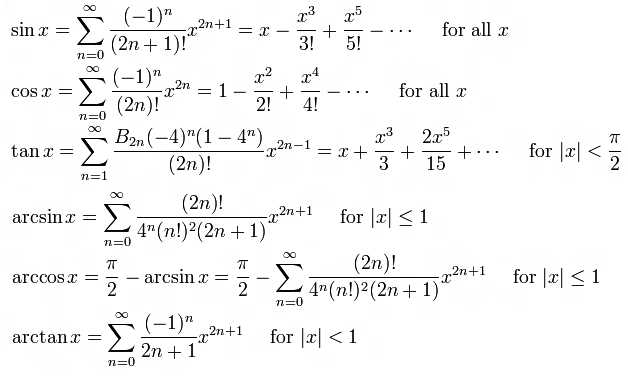


* + Công thức chuyển từ radian sang độ

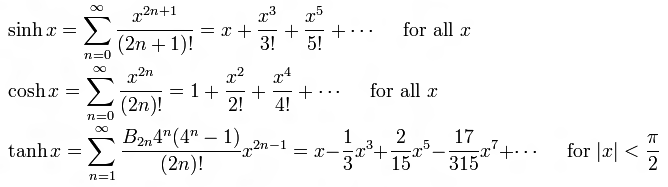
****

**Các hàm lượng giác áp dụng thuật toán chuỗi Taylor đễ tính toán :**

* + Công thức tính sin, cos ,arssin,arscos,arstan :

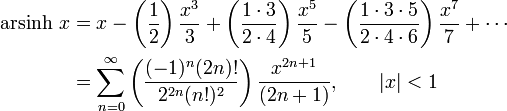
****

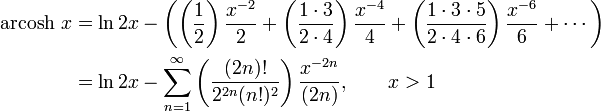
* + Công thức tính Sinh, Cosh:

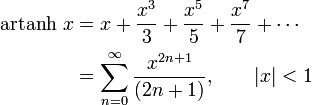


**Link:** http://en.wikipedia.org/wiki/Taylor\_series

* + Công thức tính arsSinh, arsCosh, arsTanh dựa trên công thức tính ln :







**Link:** <http://en.wikipedia.org/wiki/Inverse_hyperbolic_function>

**Tuy nhiên để dễ thực hiện coding hơn. Nhóm đã thực hiện đơn giản việc thực hiện chuỗi Taylor cho tất cả các hàm lượng giác bằng cách sử dụng các hàm lượng giác đã viết trước để thực hiện các hàm sau.**

* + Tính hàm cos

Sau khi đã tính được sin, ta dễ dàng tính được cos bằng cách cộng thêm 90o tức là : **sin(x) = cos(90-x)**

* + Tính hàm tan

Sau khi đã tính được sin và cos, ta dễ dàng tính được tan bằng công thức :

**Tan = sin/ cos**

* + Tính hàm Arccos

**Sau khi đã tính được Arcsin, ta dễ dàng tính được Arccos bằng công thức :**



**Ưu điểm :** thuật toán xữ lý với thời gian ỗn định không lâu ,các hàm lượng giác có thễ tính được số lớn.

**Khuyết điểm :** Khi chuyễn sang tính theo radian ,các hàm sin,cos,tan chỉ tính được chính xác từ 180 độ trở xuống khi áp dụng thuật toán radian. Các hàm arcSin, arcCos, arcTan, Sinh,Cosh,Tanh sẽ có sai số rất cao khi nhập số quá lớn. Các hàm Sinh,Cosh,Tanh, arc Sinh, arcCosh, arcTanh thời gian thực hiện khá lâu.

### Help.java

Tương tự như file CalculateActivity.java . Đây là class sử dụng để bắt sự kiện Button. Đồng thời class help còn có nhiệm vụ quan trọng là xữ lý sự trả về giao diên chính khi nhấn vào Button Cancel.

**Xử lý sự kiện trả về giao diện chính**

//bắt sự kiên button cancel

Button Cancel = (Button) findViewById(R.id.*cancel*);

//hàm sử lý trả về giao diên chính

Cancel.setOnClickListener(**new** Button.OnClickListener()

{

**public** **void** onClick(View v)

{

Intent intent = **new** Intent();

intent.setClass(Help.**this**, CalculateActivity.**class**);

startActivity(intent);

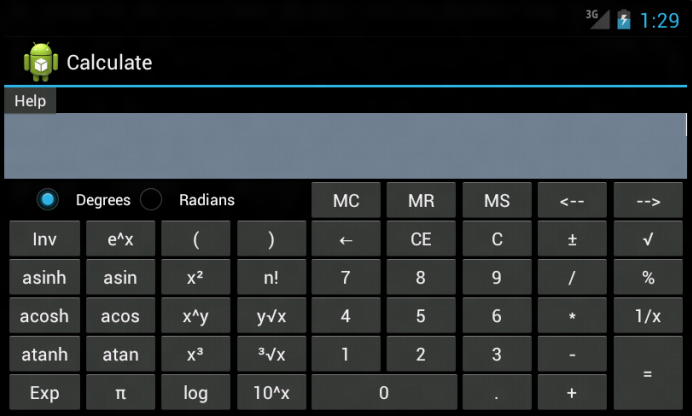
Help.**this**.finish();

}

});

# GIAO DIỆN

1. Giao diện biểu thức:



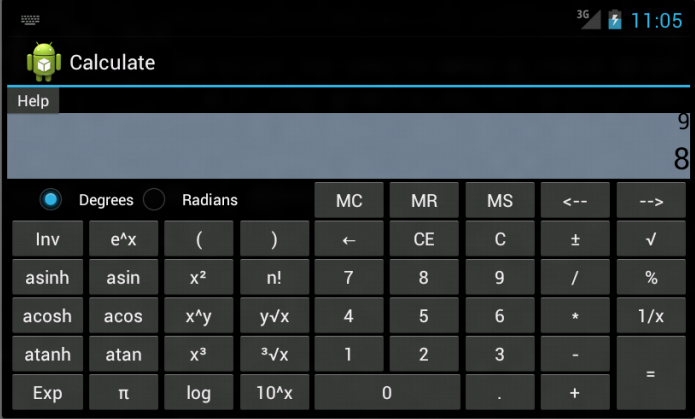
Hình 5: Giao diện biểu thức

1. Giao diện giúp đỡ khi chon button Help



**Hình 6: Giao diện thay đổi khi chọn button Help**

1. Giao diện thay đổi khi chọn button Inv:



**Hình 7: Giao diện thay đổi khi chọn button INV**

# KẾT QUẢ VÀ ĐÁNH GIÁ KẾT QUẢ

* Qua các giai đoạn phân tích, nhóm đã rút ra một kinh nghiệm đó là phải xác định và chọn công nghệ ngay từ đầu, vì nếu không xác định ngay từ đầu thì khả năng làm mới hoàn toàn lại một giai đoạn lên đến 90%.
* Nhóm đã cùng nghiên cứu Java và viết trên Eclipse. Giao diện được thiết kế bằng ngôn ngữ XML ( layout.xml). Nhưng do phải nghiên cứu cách làm mới nên nhóm mất khá nhiều thời gian, nên việc coding chương trình còn nhiều chỗ cần khắc phục.
* Đã hoàn thành được khoảng 70%.
* Cả nhóm cùng test chương trình. Chương trình còn lỗi phải khắc phục.
* Chương trình chưa hạn chế được việc nhập các số quá dài và dẫn đến việc không thể kiểm soát được
* Còn thiếu nhiều chức năng của một máy tính khoa học đầy đủ .
* Giai đoạn này hoàn thành khoảng 90%.

# KẾT LUẬN

1. Bài học kinh nghiệm:

Phải xác định và chọn công nghệ ngay từ đầu, nếu không xác định công nghệ ngay từ đầu thì khả năng làm mới hoàn toàn một giai đoạn lên tới 90%, và bị vỡ kế hoạch, gây khủng hoảng cho các thành viên trong nhóm, mất thời gian tìm hiểu tài liệu và vận dụng cái mới.

Nhóm cũng gặt gái được nhiều thành công nhất định:

* Hoàn thành được 1 phần mềm tương đối hoàn chỉnh
* Hiểu được tầm quan trọng của phân tích hệ thống.
* Học được thêm một ngôn ngữ lập trình mới.
* Kỹ năng làm việc nhóm.

1. Hướng phát triển:

* Xây dựng thêm nhiều chức năng và menu để thành 1 phần mềm máy tính đầy đủ.
* Tối ưu hóa các hàm còn chậm và tăng độ chính xác của các hàm.
* Xây dựng 1 phần mềm máy tính có khả năng in dài hoặc ngắn theo yêu cầu của người sử dụng.

TÀI LIỆU THAM KHẢO

1. http://en.wikipedia.org/wiki/Fixed-point\_arithmetic --> Operations
2. http://www.efunda.com/math/taylor\_series/taylor\_series.cfm
3. http://en.wikipedia.org/wiki/Taylor\_series
4. http://vi.wikipedia.org/wiki/H%C3%A0m\_hypebolic
5. http://developer.android.com/sdk/index.html
6. http://vietandroid.com/sa-da-ng-google-android-sdk/263-ca-i-ae-t-android-plugin-cho-netbean-ide.html
7. http://www.learn-android.com/category/beginner/
8. http://stackoverflow.com/questions/3402702/converting-floating-point-to-32-bit-fixed-point-in-java
9. http://vietandroid.com/forumdisplay.php?f=17
10. http://my.opera.com/nguyenanhkien/blog/cac-lop-can-ban-trong-java
11. Có sử dụng tham khảo đồ án của các khóa trước.