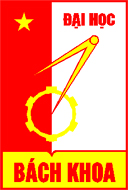
ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────



**BÀI TẬP LỚN**

HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

*(Mã học phần: IT3103)*

*Đề tài số 5:*

**Interactive simulation of composition of forces**

**Sinh viên thực hiện: NHÓM 16**

**Vũ Đức Mạnh 20225649**

**Nguyễn Đình Lượng 20225878**

**Lương Anh Minh 20225650**

**Đỗ Tuấn Minh 20225741**

**Lớp: 151965**

**Giảng viên hướng dẫn: Th.S. Nguyễn Thị Thu Trang**

**PHÂN CÔNG THÀNH VIÊN TRONG NHÓM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Công việc | Mức độ hoàn thành | Ghi chú |
| 1 | Vũ Đức Mạnh | 20225649 |  | Tốt |  |
| 2 | Nguyễn Đình Lượng | 20225878 |  | Tốt |  |
| 3 | Lương Anh Minh | 20225650 |  | Tốt |  |
| 4 | Đỗ Tuấn Minh | 20225741 |  | Tốt |  |

# MỤC LỤC

1[. LỜI CẢM ƠN](#_heading=h.2u6wntf) 4

[2. MÔ TẢ DỰ ÁN 5](#_heading=h.1mrcu09)

[2.1.](#_heading=h.46r0co2)Tổng quan dự án 5

[2.2.](#_heading=h.2lwamvv)Yêu cầu của dự án 6

[2.3.](#_heading=h.111kx3o)Biểu đồ Use Case và giải thích 8

[3: THIẾT KẾ 1](#_heading=h.3ygebqi)0

[3.1.Thiết kế lớp tổng quát 1](#_heading=h.2dlolyb)0

[3.2.Thiết kế lớp chi tiết từng Package 1](#_heading=h.sqyw64)1

[3.2.1 Package force 1](#_heading=h.3cqmetx)1

[3.2.2 Package mainObject 1](#_heading=h.1rvwp1q)2

[3.2.3 Package resources 1](#_heading=h.2r0uhxc)3

[3.2.4 Package simulator 1](#_heading=h.1664s55)3

[3.2.5 Package surface 1](#_heading=h.34g0dwd)4

[4: XÂY DỰNG CHƯƠNG TRÌNH MINH HOẠ 1](#_heading=h.25b2l0r)5

[4.1 1](#_heading=h.kgcv8k)6

[4.2 1](#_heading=h.34g0dwd)7

5.[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 1](#_heading=h.1jlao46)8

[TÀI LIỆU THAM KHẢO 1](#_heading=h.43ky6rz)9

1.LỜI CẢM ƠN

Chúng em, các thành viên của nhóm 16, xin gửi lời cảm ơn chân thành vì sự hướng dẫn và hỗ trợ nhận được trong suốt quá trình hoàn thành project "Interactivesimulation of composition of forces".

Trước tiên, chúng em xin cảm ơn giảng viên Nguyễn Thị Thu Trang, vì những lời giảng dạy tận tâm của cô và những lời khuyên sâu sắc đã giúp bọn em có thể hoàn thành project 1 cách tốt nhất.

Chúng em cũng xin cảm ơn thầy Phạm Phan Anh - trợ giảng, vì những đóng góp quý báu và sự hỗ trợ trong suốt dự án. Những phản hồi và gợi ý của anh đã đóng vai trò quan trọng trong việc cải thiện chất lượng tổng thể của công việc.

Chúng em rất vui vì có cơ hội làm việc cùng nhau trong dự án này. Cùng nhau, chúng em đã học hỏi và trưởng thành qua project này, và xin cảm ơn các bạn cùng lớp đã dành thời gian và công sức giúp chúng em giải đáp một số câu hỏi hỗ trợ cho project của chúng em.

1. **MÔ TẢ DỰ ÁN**

# Tổng quan về dự án

· **Mục tiêu:** Tạo một mô phỏng tương tác giúp người dùng hiểu về các định lý chuyển động của Newton, bao gồm các lực tác dụng lên một vật thể, cách tính toán và ảnh hưởng của các lực này đến chuyển động của vật thể.

· **Đối tượng mô phỏng:** Gồm một vật thể chính (hình khối vuông hoặc hình trụ), bề mặt ngang và một tác nhân có thể áp dụng lực lên vật thể.

· **Các tính năng chính:**

**Lựa chọn đối tượng chính:** Người dùng có thể chọn vật thể là hình khối vuông hoặc hình trụ và điều chỉnh các tham số như khối lượng, chiều dài cạnh (đối với khối vuông) hoặc bán kính (đối với hình trụ).

**Lực tác động:** Người dùng có thể điều chỉnh lực tác động theo hướng ngang, sử dụng thanh trượt hoặc nhập giá trị lực (đơn vị Newton).

**Bề mặt và ma sát:** Điều chỉnh hệ số ma sát tĩnh và động của bề mặt tác động lên vật thể.

**Mô phỏng chuyển động:** Hệ thống sẽ tính toán vị trí mới của vật thể sau mỗi khoảng thời gian, với các lực như lực hấp dẫn, lực bình thường, lực ma sát, và lực tác động từ tác nhân.

**Điều khiển mô phỏng:** Người dùng có thể tạm dừng, tiếp tục hoặc đặt lại mô phỏng, thay đổi lực và hệ số ma sát, và hiển thị thông tin chi tiết như lực, gia tốc, tốc độ, khối lượng của vật thể.

***Công cụ sử dụng***

- Ngôn ngữ lập trình: Java

- Version Control: Git (Github)

- GUI: Scene Builder

# 2.2 Yêu cầu của dự án

Dự án yêu cầu xây dựng một chương trình mô phỏng tương tác về sự kết hợp của các lực tác động lên một vật thể, dựa trên các định lý chuyển động của Newton. Chương trình này sẽ giúp người dùng hiểu và trực quan hóa các lực cơ bản như lực hấp dẫn, lực bình thường, lực ma sát và lực tác động ngang trên một vật thể. Mô phỏng sẽ cho phép người dùng điều khiển các yếu tố của hệ thống vật lý và quan sát chuyển động của vật thể dưới sự tác động của các lực này.

### ****a.Giao diện người dùng (GUI)****

**Đối tượng chính:** Người dùng có thể chọn vật thể là hình khối vuông hoặc hình trụ.

**Lựa chọn tham số của vật thể:**

Đối với vật thể hình khối vuông: Người dùng có thể điều chỉnh chiều dài cạnh và khối lượng.

Đối với vật thể hình trụ: Người dùng có thể điều chỉnh bán kính và khối lượng.

**Tác nhân (Actor):** Người dùng có thể điều khiển lực tác động lên vật thể bằng cách sử dụng thanh trượt hoặc nhập giá trị lực (đơn vị Newton).

**Ma sát:** Điều chỉnh hệ số ma sát tĩnh và động của bề mặt tác động lên vật thể qua thanh trượt và ô nhập liệu. Lưu ý rằng hệ số ma sát tĩnh phải lớn hơn hệ số ma sát động.

**Hiển thị thông tin:** Hiển thị các thông số như lực, tổng lực, khối lượng, tốc độ và gia tốc của vật thể. Cung cấp tùy chọn hiển thị hoặc ẩn các thông tin chi tiết này qua các hộp kiểm.

### ****b.Mô phỏng chuyển động****

**Cập nhật vị trí và vận tốc:** Sau mỗi khoảng thời gian Δt, chương trình cần tính toán và cập nhật vị trí và vận tốc của vật thể dựa trên các lực tác động.

**Chuyển động quay (cho vật thể hình trụ):** Đối với vật thể hình trụ, chương trình cần tính toán chuyển động quay bổ sung, sử dụng các công thức vật lý phù hợp với chuyển động quay.

**Công thức chuyển động:**

Tính gia tốc, vận tốc và vị trí của vật thể theo công thức của Newton.

Đối với vật thể hình trụ, tính cả chuyển động quay và vận tốc góc.

### ****c.Các yếu tố vật lý và công thức tính toán****

**Lực hấp dẫn:** Lực này luôn tác dụng theo phương thẳng đứng, hướng xuống, có giá trị bằng khối lượng của vật thể nhân với gia tốc trọng trường (g = 10 m/s²).

**Lực bình thường:** Lực này tác dụng theo phương thẳng đứng, hướng lên, và có giá trị bằng lực hấp dẫn.

**Lực ma sát:** Lực này có hướng ngược lại với hướng chuyển động của vật thể và phụ thuộc vào hệ số ma sát và lực tác động.

**Lực tác động:** Lực này luôn được áp dụng vào trọng tâm của vật thể theo hướng ngang và có thể điều chỉnh trong mô phỏng.

# 2.3 Biểu đồ Use case và giải thích

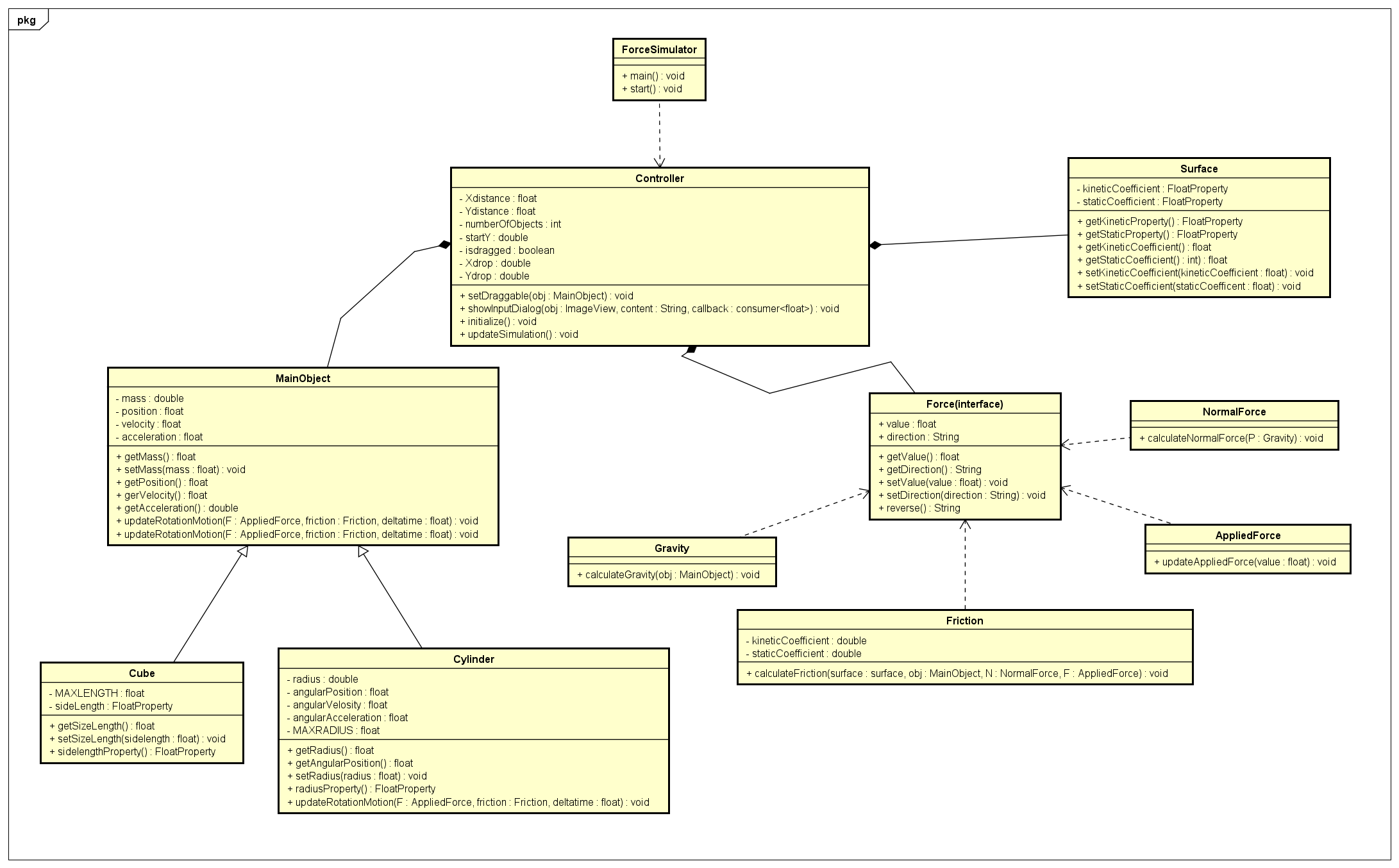
# UseCase Diagram0 (1)

**Giải Thích:**

* **Animate background:** Người dùng có thể chọn hiệu ứng hoạt hình cho nền của mô phỏng.
* **Select Object Type:** Người dùng lựa chọn loại vật thể muốn mô phỏng (ví dụ: quả bóng, hộp, v.v.).
* **Apply Force:** Người dùng áp dụng lực lên vật thể đã chọn.
* **Adjust Coefficients:** Người dùng điều chỉnh các hệ số vật lý như ma sát, trọng lực, đàn hồi để thay đổi hành vi của vật thể.
* **Observe motion:** Người dùng quan sát chuyển động của vật thể dưới tác động của lực.
* **Start simulation:** Người dùng bắt đầu quá trình mô phỏng.
* **Pause/Resume Simulation:** Người dùng tạm dừng hoặc tiếp tục quá trình mô phỏng.
* **Reset Simulation:** Người dùng đặt lại mô phỏng về trạng thái ban đầu.
* **View Simulation Data:** Người dùng xem dữ liệu chi tiết về quá trình mô phỏng (ví dụ: vị trí, vận tốc, lực tác dụng).
* **Drag object:** Người dùng trực tiếp kéo vật thể trên màn hình để thay đổi vị trí ban đầu.

# 3. THIẾT KẾ

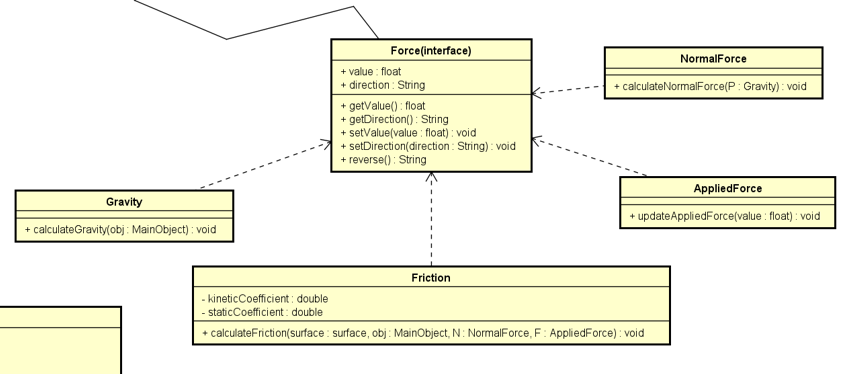
**3.1.Thiết kế lớp chung**



**3.2.Thiết kế lớp chi tiết từng Package**

## 3.2.1 Package force

### Package force chứa các đối tượng đại diện cho lực tác động và ma sát trong quá trình mô phỏng chuyển động vật lý. Các lớp này thực hiện vai trò tính toán các lực tác động lên các đối tượng trong hệ thống.



### Lớp ****Force****

· **Thuộc tính:**

·  **value:** Một đối tượng Float Property , lưu trữ giá trị của lực.

· **Phương thức:**

**getValue()**: Trả về giá trị của lực dưới dạng **Float**

**getValueProperty()**: Trả về đối tượng **Float Property** của lực, cho phép theo dõi sự thay đổi của giá trị lực.

**setValue(float value)**: Cập nhật giá trị của lực.

### Lớp ****AppliedForce**** (Kế thừa từ lớp Force)

· **Thuộc tính:**

· Kế thừa thuộc tính value từ lớp Force

· **Phương thức:**

· **setValue(float value)**: Cập nhật giá trị lực áp dụng, gọi phương thức setValue( ) từ lớp cha (Force).

### Lớp ****Friction**** (Kế thừa từ lớp Force)

· **Thuộc tính:**

· Kế thừa thuộc tính value từ lớp Force

· **Phương thức:**

· **calculateFriction(Surface surface, MainObject obj, NormalForce N, AppliedForce F)**: Tính toán lực ma sát dựa trên các tham số đầu vào như bề mặt, đối tượng, lực pháp tuyến và lực áp dụng. Lực ma sát được tính toán theo các điều kiện khác nhau (tĩnh và động) và dựa trên các loại đối tượng (Cube hoặc Cylinder).

### Lớp ****Gravity**** (Kế thừa từ lớp Force)

· **Thuộc tính:**

· Kế thừa thuộc tính value từ lớp Force

· **Phương thức:**

· **calculateGravity(MainObject obj)**: Tính toán trọng lực của đối tượng dựa trên khối lượng của nó (trọng lực = 10 \* khối lượng).

.

### Lớp ****NormalForce**** (Kế thừa từ lớp Force)

· **Thuộc tính:**

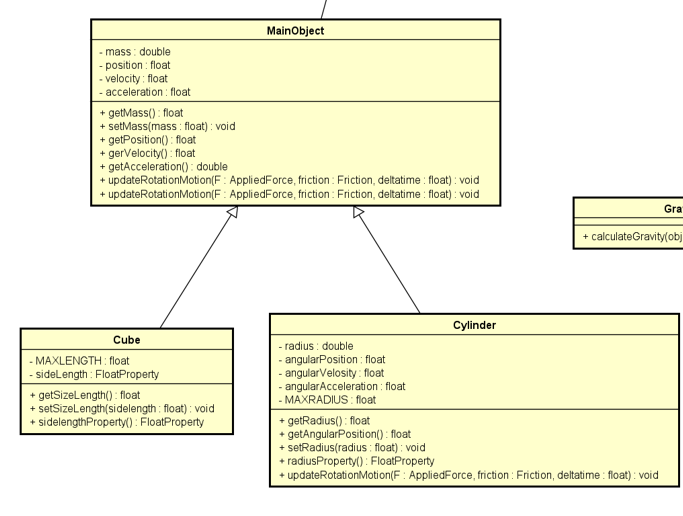
· Kế thừa thuộc tính value từ lớp Force

· **Phương thức:**

· **calculateNormalForce(Gravity P)**: Cập nhật lực pháp tuyến (Normal Force) bằng giá trị trọng lực.

### 3.2.2 Package mainObject

### Package mainObject chứa các đối tượng đại diện cho các đối tượng vật lý như mainObject, Cube, và Cylinder. Các đối tượng này thực hiện quá trình mô phỏng lực và chuyển động vật lý trong chương trình. Các đối tượng sẽ sử dụng lực tác động và ma sát để tính toán chuyển động dịch chuyển và quay của chúng.

****

#### **Lớp MainObject**

**Thuộc tính:**

* Mass (FloatProperty): Khối lượng của đối tượng.
* position (FloatProperty): Vị trí của đối tượng.
* velocity (FloatProperty): Vận tốc của đối tượng.
* acceleration (FloatProperty): Gia tốc của đối tượng.

**Phương thức:**

* **getMass()**: Trả về khối lượng của đối tượng.
* **setMass(float mass)**: Thiết lập khối lượng cho đối tượng.
* **getPosition()**: Trả về vị trí của đối tượng.
* **setPosition(float position)**: Thiết lập vị trí cho đối tượng.
* **getVelocity()**: Trả về vận tốc của đối tượng.
* **setVelocity(float velocity)**: Thiết lập vận tốc cho đối tượng.
* **getAcceleration()**: Trả về gia tốc của đối tượng.
* **setAcceleration(float acceleration)**: Thiết lập gia tốc cho đối tượng.
* **updateTranslationMotion(AppliedForce F, Friction friction, float deltatime)**: Cập nhật chuyển động dịch chuyển của đối tượng.
* **updateRotationMotion(AppliedForce F, Friction friction, float deltatime)**: Phương thức trừu tượng cho chuyển động quay, cần được cài đặt trong các lớp con.

#### **Lớp Cube**

**Thuộc tính:**

* sideLength (FloatProperty): Độ dài cạnh của khối lập phương (cube).

**Phương thức:**

* **getSizeLength()**: Trả về độ dài cạnh của Cube.
* **setSizeLength(float sideLength)**: Thiết lập độ dài cạnh cho Cube.
* **updateTranslationMotion(AppliedForce F, Friction friction, float deltatime)**: Cập nhật chuyển động dịch chuyển cho Cube.
* **updateRotationMotion(AppliedForce F, Friction friction, float deltatime)**: Không có chuyển động quay trong Cube, để trống.

#### **Lớp Cylinder**

**Thuộc tính:**

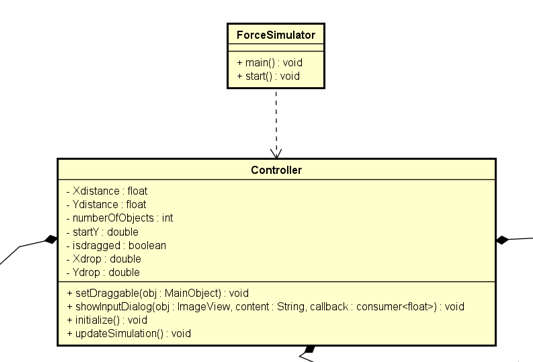
* **radius (FloatProperty)**: Bán kính của hình trụ (cylinder).
* **angularPosition (FloatProperty)**: Vị trí góc của hình trụ.
* **angularVelocity (FloatProperty)**: Tốc độ góc của hình trụ.
* **angularAcceleration (FloatProperty)**: Gia tốc góc của hình trụ.

**Phương thức:**

* **getRadius()**: Trả về bán kính của Cylinder.
* **setRadius(float radius)**: Thiết lập bán kính cho Cylinder.
* **updateTranslationMotion(AppliedForce F, Friction friction, float deltatime)**: Cập nhật chuyển động dịch chuyển cho Cylinder.
* **updateRotationMotion(AppliedForce F, Friction friction, float deltatime)**: Cập nhật chuyển động quay cho Cylinder.

### 3.2.3 Package simulator

Package simulator chứa các lớp điều khiển quá trình mô phỏng chính. Gói này có thể thực hiện các bước như bắt đầu mô phỏng, cập nhật trạng thái của các đối tượng (Cube, Cylinder), tính toán lực tác động và ma sát, và thực hiện vòng lặp mô phỏng.



**Lớp Controller:**

**Thuộc tính:**

· **Xdistance, Ydistance**: Lưu khoảng cách từ chuột đến đối tượng để di chuyển mượt.

· **numberOfObjects**: Theo dõi số lượng đối tượng để quản lý thêm mới.

· **startY**: Lưu vị trí Y ban đầu để kiểm tra di chuyển.

· **isDragged**: Xác định đối tượng có đang bị kéo.

· **Xdrop, Ydrop**: Lưu vị trí thả đối tượng cuối cùng.

**Phương thức:**

· **showInputDialog**: Hiển thị hộp thoại nhập liệu để người dùng cung cấp thông tin, ví dụ như các tham số lực.

· **setDraggable**: Cho phép người dùng kéo và di chuyển đối tượng trên giao diện bằng cách gắn các sự kiện chuột.

· **initialize()**: Khởi tạo các thành phần giao diện và cấu hình ban đầu cho ứng dụng.

· **updateSimulation**: Cập nhật trạng thái và vị trí của các đối tượng trong mô phỏng theo logic vật lý.

**Lớp Force Simulator:**

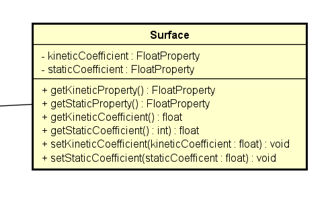
### Thuộc tính

### Phương thức

* **main(String[] args)**: Khởi chạy ứng dụng với launch(args)
* **start(Stage primaryStage)**: Tải FXML, tạo Scene, gắn vào primaryStage
* , hiển thị cửa sổ.

### 3.2.4 Package surface

Package surface chứa lớp surface quản lý các hệ số ma sát (động và tĩnh), cung cấp các phương thức để lấy và thay đổi các giá trị này thông qua các đối tượng FloatProperty của JavaFX, cho phép theo dõi sự thay đổi giá trị và cập nhật giao diện người dùng nếu có.



### Thuộc tính:

* **kineticCoefficient**: Thuộc tính kiểu FloatProperty, lưu giá trị hệ số ma sát động, khởi tạo giá trị mặc định là 0.
* **staticCoefficient:** Thuộc tính kiểu FloatProperty, lưu giá trị hệ số ma sát tĩnh, khởi tạo giá trị mặc định là 0.

### Phương thức:

* **getKineticProperty()**: Trả về đối tượng FloatProperty của hệ số ma sát động.
* **getStaticProperty()**: Trả về đối tượng FloatProperty của hệ số ma sát tĩnh.
* **getKineticCoefficient()**: Trả về giá trị hệ số ma sát động dưới dạng float.
* **getStaticCoefficient()**: Trả về giá trị hệ số ma sát tĩnh dưới dạng float.
* **setKineticCoefficient(float kineticCoefficient):** Cập nhật giá trị hệ số ma sát động.
* **setStaticCoefficient(float staticCoefficient)**: Cập nhật giá trị hệ số ma sát tĩnh

.