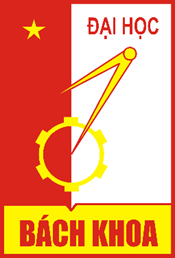
**Hanoi University of Science and Technology**

****

**OOP & Java Lab**

**Report for Mini-Project**

**Team 06**

**Project Name**: 7. Demonstration of types of COVID-19 virus and its mechanism

# Members:

# Bùi Trần Hải Quân – 20194821

# Nguyễn Minh Quân – 20194823

# Trần Quang Thái – 20194836

# Assignment of members

# Bui Tran Hai Quan

# Search information for virus about: structure of each specific virus, the infection stages, symptoms, and infecting method of all viruses.

# Collect images about structure of virus.

# hust.soict.globalict.Virus: Herpes, COVID-19, Rota, Astro, Adeno

# hust.soict.globalict.Assets: Herpes, COVID-19, Rota, Astro, Adeno

# Initial data for constructor of virus in [hust](https://github.com/tqthai2001/OOLT.ICT.20212.Team06/tree/main/Virus/src/hust)/[soict](https://github.com/tqthai2001/OOLT.ICT.20212.Team06/tree/main/Virus/src/hust/soict)/globalict/VirusMain

# MainScreen and HelpScreen

# Nguyen Minh Quan

# Implement the types of Viruses: Enveloped and Non-Enveloped

# Build attributes of Virus

# Construct the Structure Screen of all viruses

# hust.soict.globalict.Virus: Element, VirusWith/WithoutEnvelope

# [hust](https://github.com/tqthai2001/OOLT.ICT.20212.Team06/tree/feature/attack/Virus/src/hust)/[soict](https://github.com/tqthai2001/OOLT.ICT.20212.Team06/tree/feature/attack/Virus/src/hust/soict)/[globalict](https://github.com/tqthai2001/OOLT.ICT.20212.Team06/tree/feature/attack/Virus/src/hust/soict/globalict)/Screen: VirusStructureScreen

# Initially implements the VirusMain

# hust.soict.globalict.Controller: VirusStructureController

# Tran Quang Thai

# Implement screen of all infecting stages for virus: Virus With Envelope, Virus Without Envelope

# [hust](https://github.com/tqthai2001/OOLT.ICT.20212.Team06/tree/feature/attack/Virus/src/hust)/[soict](https://github.com/tqthai2001/OOLT.ICT.20212.Team06/tree/feature/attack/Virus/src/hust/soict)/[globalict](https://github.com/tqthai2001/OOLT.ICT.20212.Team06/tree/feature/attack/Virus/src/hust/soict/globalict)/Screen: Infecting Screen, Infecting Detail Screen

# hust.soict.globalict.Controller: Virus Controller, Virus Infecting Controller, Virus Infecting Detail Controller

# Merge and Fix all work of each member

# Adding attributes and refactor structure of Virus

# Design the Relationship: Composition & Aggregation

# Design Polymorphism

# Mini-project Description

# Mini-project requirement

# Project Name: Demonstration of types of COVID-19 virus and its mechanism

# The main purpose of this project is to illustrate the detail structure and the infecting stages of some common viruses. As we all know, COVID-19 pandemic has been affected deeply to our lives all over the world: fallen-down economy, millions of people has died,… Thus, there is a necessary requirement of understanding the different types of the virus, as well as the way they infect to have the basic knowledge to prevent them. Every virus has 2 basic elements: acid nucleic and capsid. Based on their structure, viruses are divided into 2 categories: Enveloped virus and Non-enveloped virus. Our application, which contains 6 viruses (HIV, COVID-19, Herpes, Rota, Astro and Adeno virus), will concentrate on what the viruses are constructed from and how they attack host-cells.

# Use-case diagram and explaination

# Diagram Description automatically generated

# Firstly, users could have ability to access the application. In the main menu screen, they can click at “Help” button to get the support of introducing about this project and instructions for use. Users can also pick up the type of virus: “Virus with envelope” or “Virus without envelope”. “Virus with envelope” contains HIV, COVID-19, Herpes. Meanwhile, “Virus without envelope” includes Rota, Astro and Adeno virus. Users can observe the particular structure of each virus if they choose one type of above viruses. After that, they can view the infecting stages step by step. Moreover, this application has a “go back” area, which makes returning to main menu convenient.

# Design

# General class diagram

# Diagram Description automatically generated

# We can see from the general class diagram that this project is followed by the MVC model. Package Controller helps executing the package Screen and the application be smoother.

# In the package Virus, class VirusEnvelope and class VirusNonEnvelope extend class Virus, because every virus in our knowledge could be divided into these 2 types, they inherit all common characteristics of general viruses.

# Virus class has aggregation relationship with Capsid class and Acid Nucleic because all viruses must have these 2 elements but when the virus attack host-cell, virus took away capsid and re-construct the Acid, which means they are Has-a relation and independent.

# VirusEnvelope has composition relationship with Envelope class because envelope is compulsory for us to discriminate between Enveloped virus and Non-enveloped virus. Enveloped virus can’t live without their “shell”.

# Packages and detail class diagrams

# Virus class

# Text, email Description automatically generated

# Element package

# Diagram Description automatically generated

# Element is a class to describe every biological part of a virus. Thus, AcidNucleic, Capsid, Envelope and OtherElement classes extend from Element class

# VirusEnvelope package

# A picture containing text Description automatically generated

# HIV, COVID-19, Herpes are Enveloped virus, so they must extend VirusWithEnvelope.

# VirusNonEnvelope package

# A picture containing diagram Description automatically generated

# Astro, Rota and Adeno are all Non-enveloped virus. They extend from VirusWithoutEnvelope.

# Screen package

# Diagram Description automatically generated

# Controller package

# Diagram Description automatically generated

# OOP design:

# Inheritance: Enveloped Virus, Non-enveloped virus inherit Virus class; HIV, COVID-19, Herpes inherit Enveloped Virus; Rota, Astro, Adeno inherit Non-Enveloped Virus …

# Encapsulation: method getters, setters in each virus class: getTegument(); getmProtein(); getFiber();…

# Abstraction: abstracted class Element: describe biological components of a virus

# Polymorphism: getDetail() method; Downcasting from Virus class to VirusWithEnvelope, infect() method