

# Temat projektu: Symulacja wioski

Jakub Tadeusz - 256760  
Michał Szymański - 252768

June 10, 2020

## Contents

<b>1</b>	<b>Szczegółowy opis systemu</b>	<b>1</b>
1.1	Analiza czasownikowo-rzeczownikowa . . . . .	1
1.2	Diagramy CRC . . . . .	1
1.3	Diagram przypadków użycia . . . . .	5
1.4	Diagramy klas . . . . .	6
<b>2</b>	<b>Logika symulacji</b>	<b>8</b>
2.1	Diagramy aktywności . . . . .	8

## 1 Szczegółowy opis systemu

### 1.1 Analiza czasownikowo-rzeczownikowa

Główna klasa symulacji posiada informacje dotyczące stanu wioski (zaludnienie, dostatek, zdolność bojowa). Na początku każdego dnia, składająca się z mieszkańców rada wioski wybiera poprzez głosowanie zadania, po czym generowane są zdarzenia losowe. W przypadku spowodowanego wydarzeniami spadku ludności do zera, symulacja kończy się, w przeciwnym razie uruchamiany jest kolejny dzień.

### 1.2 Diagramy CRC

<b>Classname:</b>	Village
<b>Superclass:</b>	none
<b>Subclass(es):</b>	none
<b>Responsibilities:</b> Checks if village state is ok Stores village properties	<b>Colaboration:</b> Simulation SimulationFactory BattleEvent Event

<b>Classname:</b>	SimulationConsts
<b>Superclass:</b>	none
<b>Subclass(es):</b>	none
<b>Responsibilities:</b> Stores parameters which are used to randomize Village's fluctuations Stores default IO config	<b>Colaboration:</b> BattleEvent RandomEvent RandomEventGenerator ArgumentParser VillageCouncil SeletcedEvent

<b>Classname:</b>	Simulation
<b>Superclass:</b>	none
<b>Subclass(es):</b>	none
<b>Responsibilities:</b> Handles simulation logic Establishes actual difficulty Prints daily summary Executes events	<b>Colaboration:</b> Village VillageCouncil RandomEventGenerator Event VillageCouncil SimulationFactory App

<b>Classname:</b>	VillageCouncil
<b>Superclass:</b>	none
<b>Subclass(es):</b>	none
<b>Responsibilities:</b> Adds new council members Generates council members Gets council memebers votes which are used to decide which event should apply	<b>Colaboration:</b> Simulation CouncilMember

<b>Classname:</b>	CouncilMember
<b>Superclass:</b>	none
<b>Subclass(es):</b>	none
<b>Responsibilities:</b> Gets member choice Rerolls member	<b>Colaboration:</b> VillageCouncil

<b>Classname:</b>	Event
<b>Superclass:</b>	none
<b>Subclass(es):</b>	SelectedEvent BattleEvent RandomEvent
<b>Responsibilities:</b> Executes event effects on Village Stores changes to apply	<b>Colaboration:</b> none

<b>Classname:</b>	SelectedEvent
<b>Superclass:</b>	Event
<b>Subclass(es):</b>	none
<b>Responsibilities:</b> Calculates changes from selected event due to difficulty	<b>Colaboration:</b> VillageCouncil Event SimulationConsts

<b>Classname:</b>	BattleEvent
<b>Superclass:</b>	Event
<b>Subclass(es):</b>	none
<b>Responsibilities:</b> Gets oponent to battle Initiate battle Executes changes as a result of battle	<b>Colaboration:</b> Village Event SimulationConsts RandomEventGenerator VillageCouncil

<b>Classname:</b>	RandomEventGenerator
<b>Superclass:</b> <b>Subclass(es):</b>	none none
<b>Responsibilities:</b> Randomize event from available event list Returns randomized event	<b>Colaboration:</b> RandomEvent BattleEvent Event SimulationConsts SimulationFactory Simulation

<b>Classname:</b>	RandomEvent
<b>Superclass:</b> <b>Subclass(es):</b>	none none
<b>Responsibilities:</b> Sets Random changes	<b>Colaboration:</b> RandomEventGenerator Event SimulationConsts

<b>Classname:</b>	SimulationFactory
<b>Superclass:</b> <b>Subclass(es):</b>	none none
<b>Responsibilities:</b> Creates statring village and simulation Sets actual IO types	<b>Colaboration:</b> Village RandomEventGenerator Simulation App

<b>Classname:</b>	App
<b>Superclass:</b> <b>Subclass(es):</b>	none none
<b>Responsibilities:</b> Has a main method which starts whole simulation	<b>Colaboration:</b> Simulation SimulationFactory ArgumentParser

<b>Classname:</b>	ArgumentParser
<b>Superclass:</b> <b>Subclass(es):</b>	none none
<b>Responsibilities:</b> Decides which IO type application should use	<b>Colaboration:</b> SimulationConsts ConsoleOutput FileOutput ConsoleInput FileInput DefaultInput

<b>Classname:</b>	DefaultInput
<b>Superclass:</b> <b>Subclass(es):</b>	none none
<b>Responsibilities:</b> Has methods used to load starting properties when user did not pass them	<b>Colaboration:</b> ArgumentParser

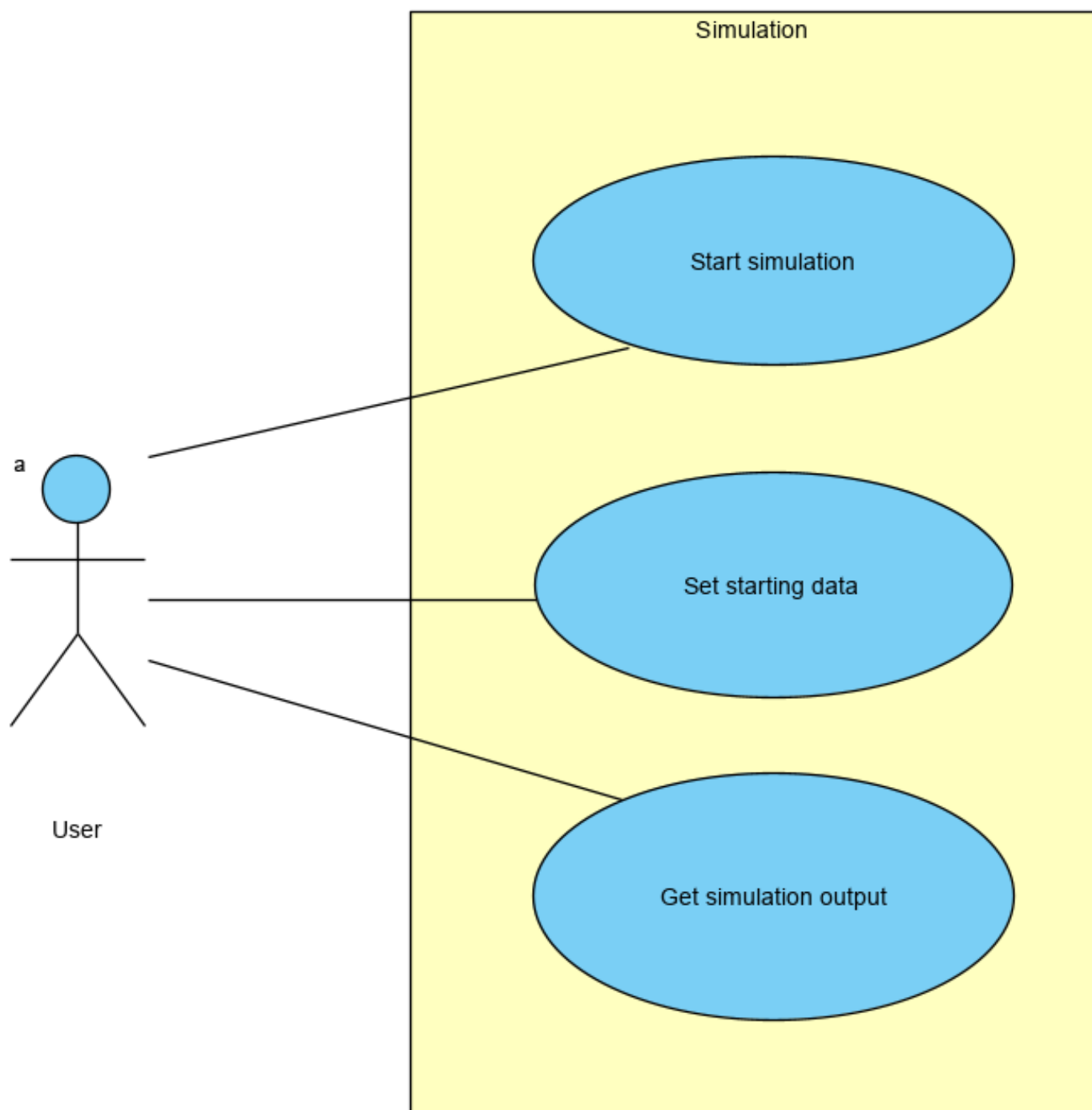
<b>Classname:</b>	FileOutput
<b>Superclass:</b>	none
<b>Subclass(es):</b>	none
<b>Responsibilities:</b> Has methods used to print simulation logs into file	<b>Colaboration:</b> ArgumentParser

<b>Classname:</b>	FileInput
<b>Superclass:</b>	none
<b>Subclass(es):</b>	none
<b>Responsibilities:</b> Has methods used to scan starting village properties from file	<b>Colaboration:</b> ArgumentParser

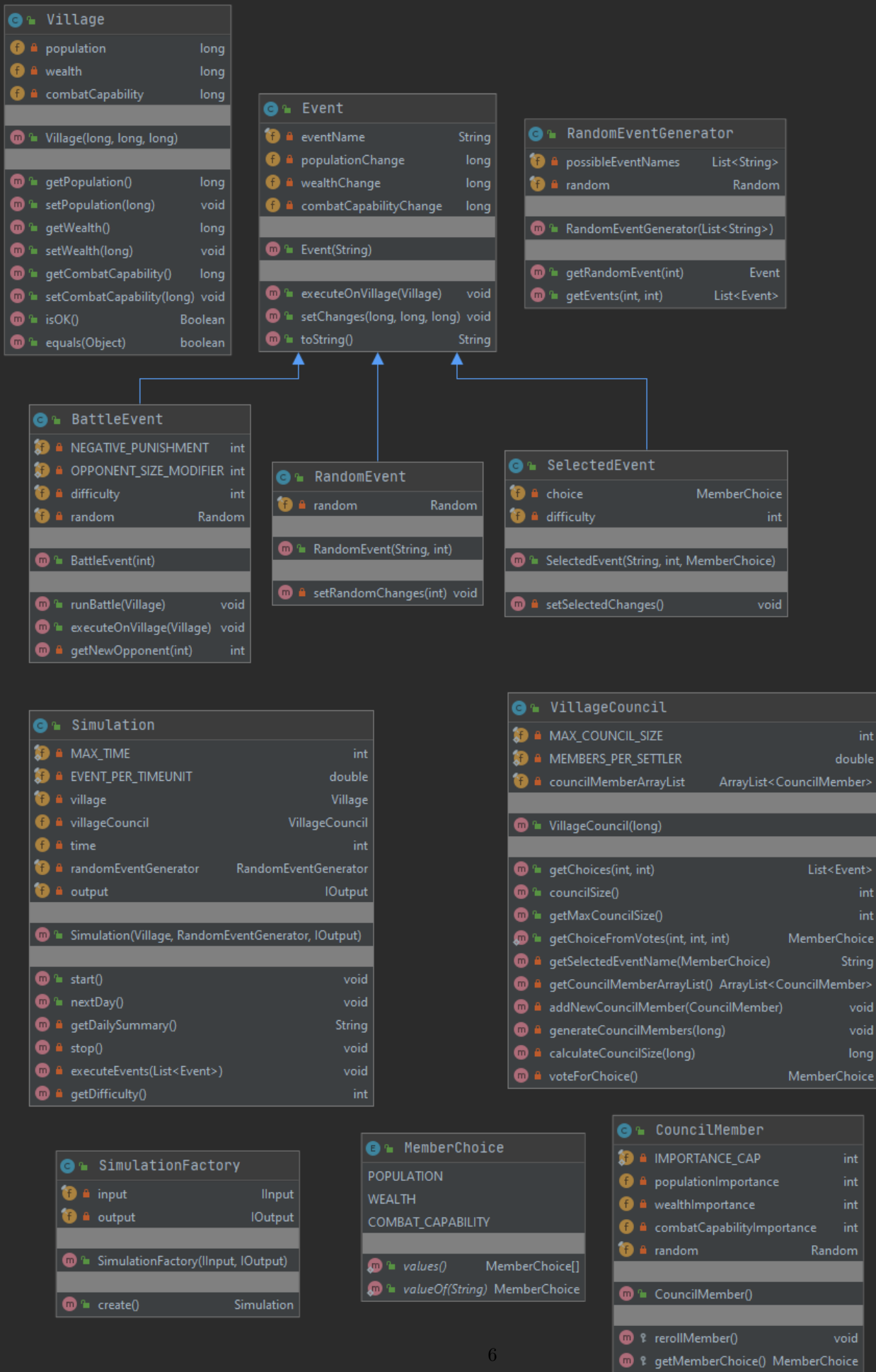
<b>Classname:</b>	ConsoleInput
<b>Superclass:</b>	none
<b>Subclass(es):</b>	none
<b>Responsibilities:</b> Has methods used to scan starting village properties from console	<b>Colaboration:</b> ArgumentParser

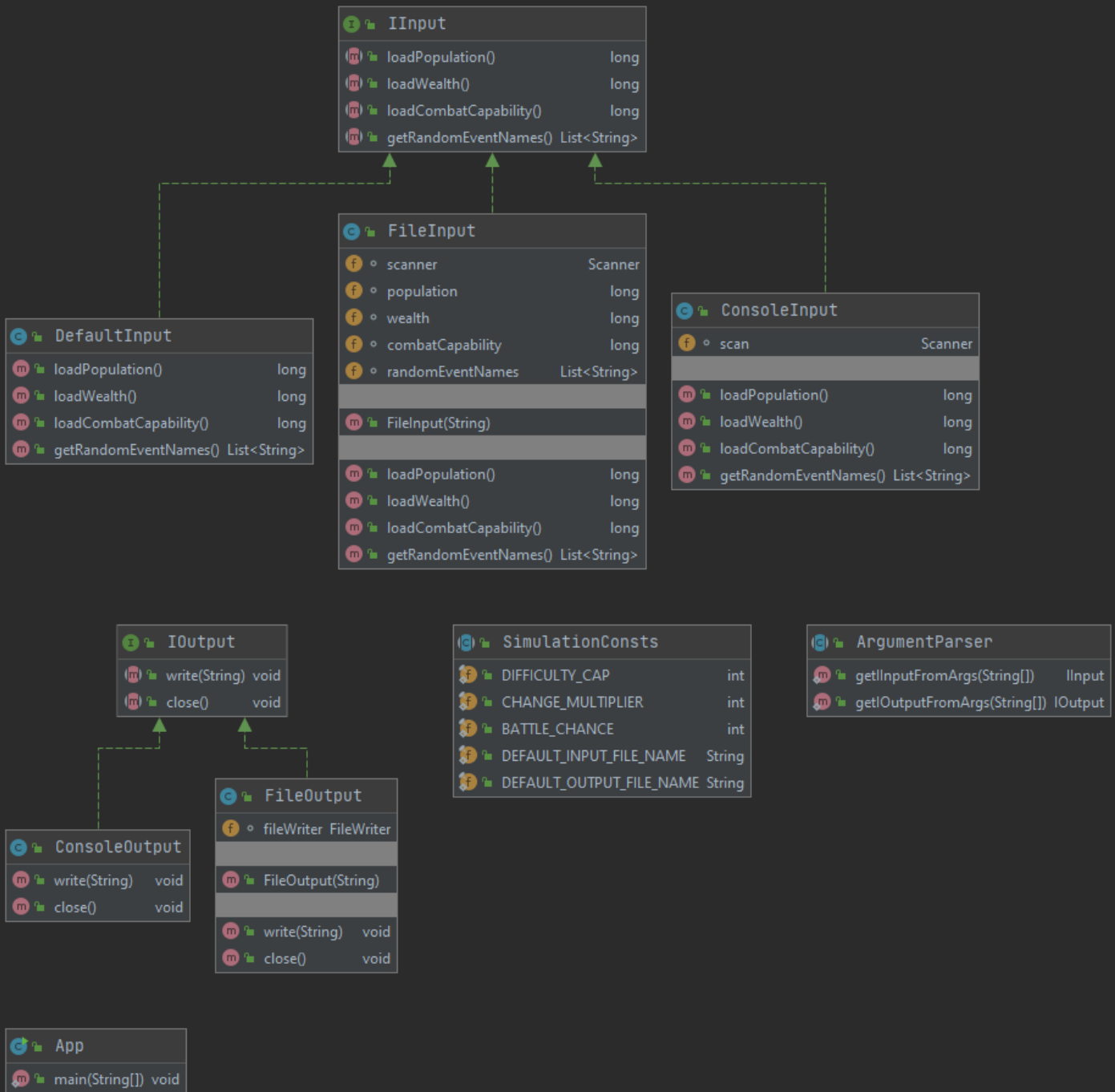
<b>Classname:</b>	ConsoleOutput
<b>Superclass:</b>	none
<b>Subclass(es):</b>	none
<b>Responsibilities:</b> Has methods used to print simulation logs into console	<b>Colaboration:</b> ArgumentParser

### 1.3 Diagram przypadków użycia



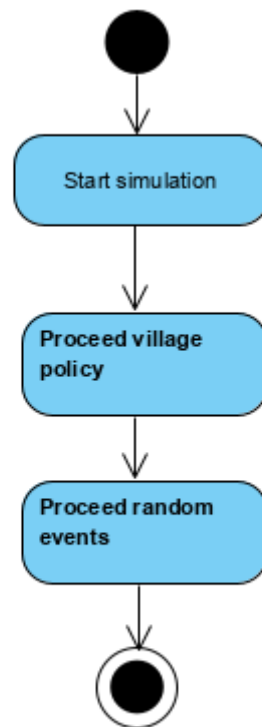
## 1.4 Diagramy klas





## 2 Logika symulacji

### 2.1 Diagramy aktywności



Proceed Village policy

