Programowanie Obiektowe – Projekt w grupach

**Dokumentacja**

# ***Wstęp***

* Zajęcia i prowadzący:

- Wtorek, 18:55

- Mgr inż. Paweł Majewski

* Skład grupy:

- Michał Dziedziak 263901 (lider)

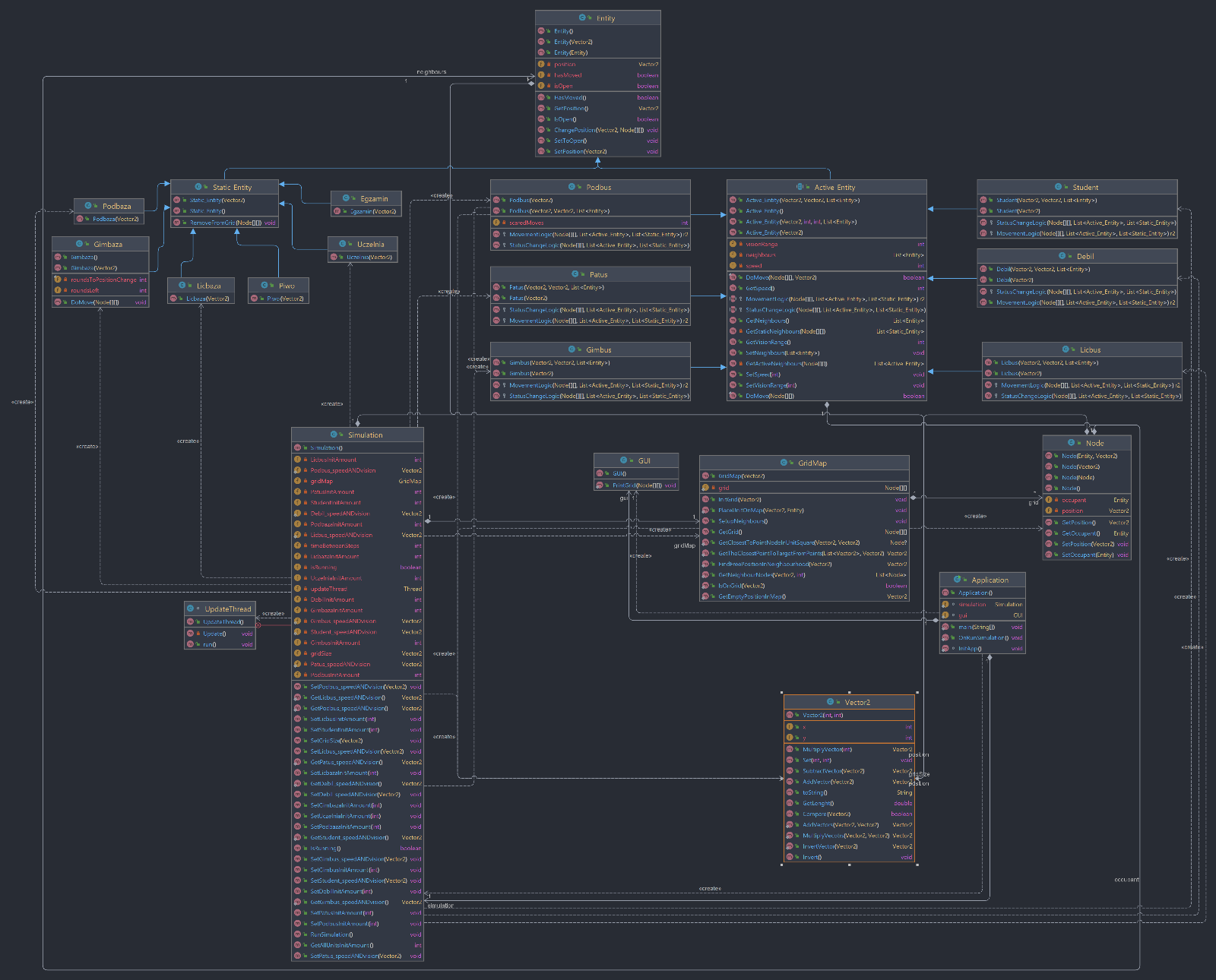
- Karol Waszkiewicz 263918

* Opis projektu:

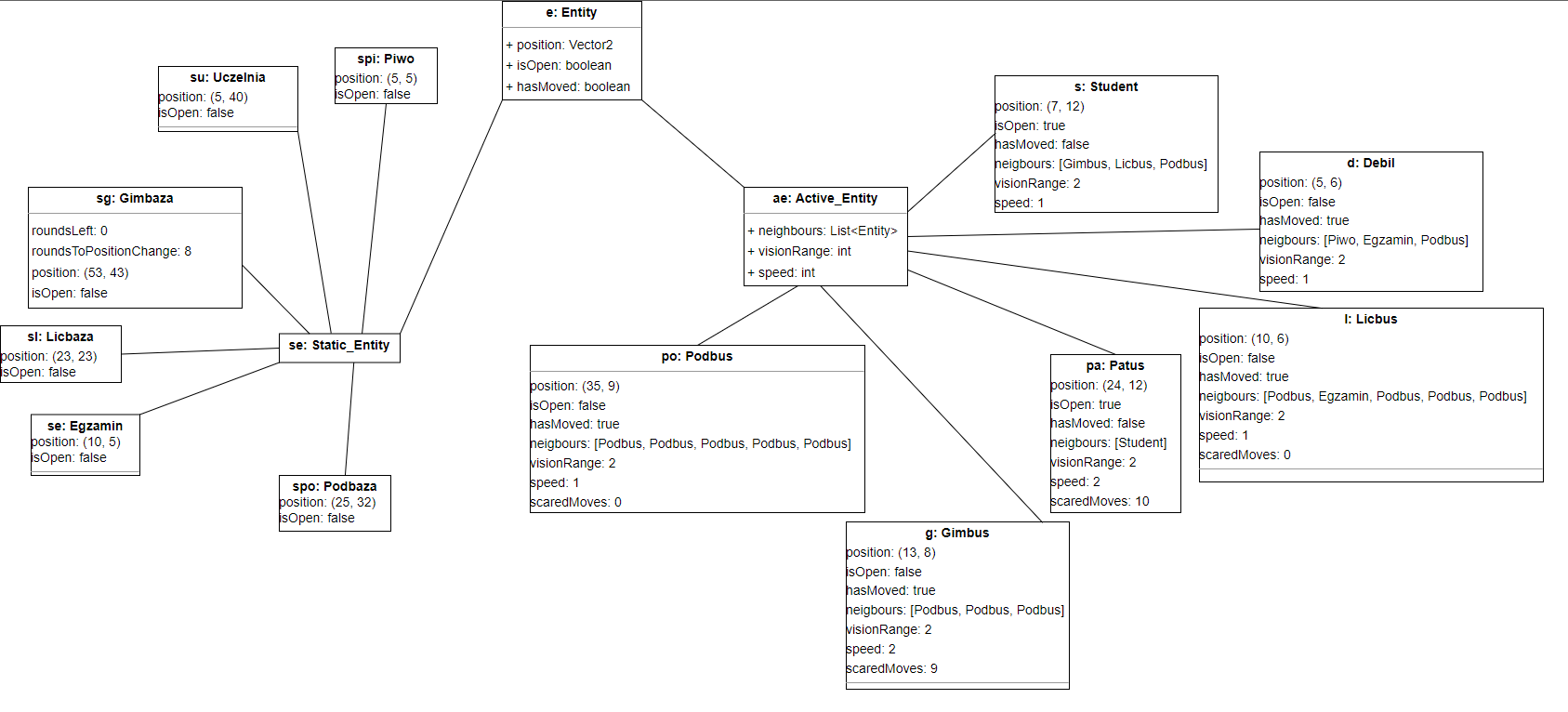
Projekt ten to aplikacja symulująca rozwój społeczności, w której każdy członek musi przejść przez pewien cykl. Celem samej symulacji jest dojście przez pewnego członka do końca cyklu, co ją zatrzymuje. Na prostokątnej planszy umiejscowione są dwa rodzaje bytów: ruchome (wcześniej wspomniani członkowie) oraz nieruchome. Zadaniem tych drugich jest umożliwienie kontynuacji symulacji, gdyż dzięki nim członkowie mogą awansować dalej, w głąb cyklu. Same byty nieruchome także dzielą się na takie, które istnieją od początku symulacji oraz na takie, które mogą pojawiać się i znikać przy pewnych okolicznościach. Wracając do bytów ruchomych, każdy z nich ma specjalne właściwości, co w pewien sposób może ułatwiać bądź utrudniać zakończenie symulacji. Dlatego też symulacja taka w dużej mierze oparta jest na losowości i generatorze liczb losowych – jej długości trwania mogą się diametralnie od siebie różnić.

# ***Diagramy***

* Diagram klas:

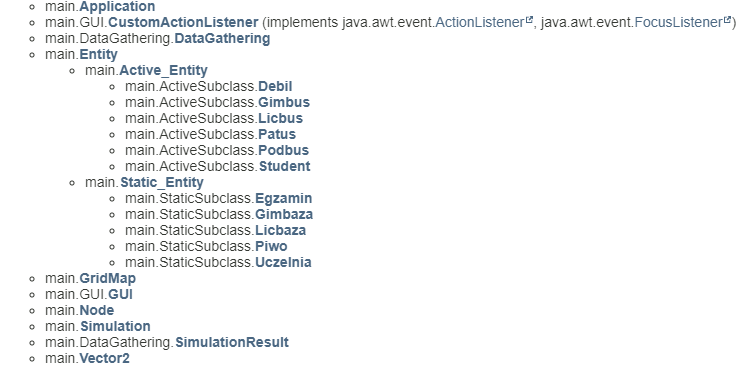


* Diagram obiektów:

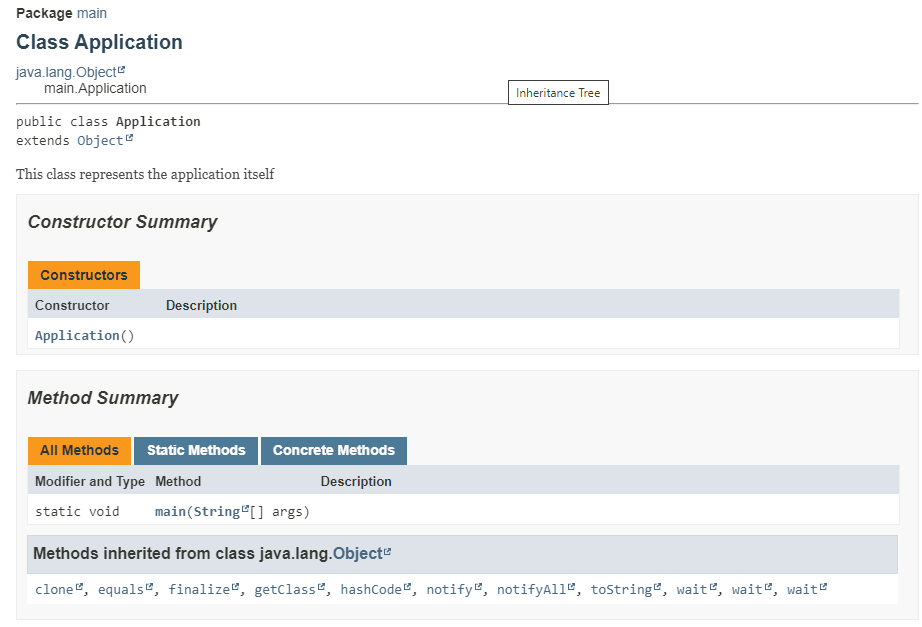


* JavaDocs:

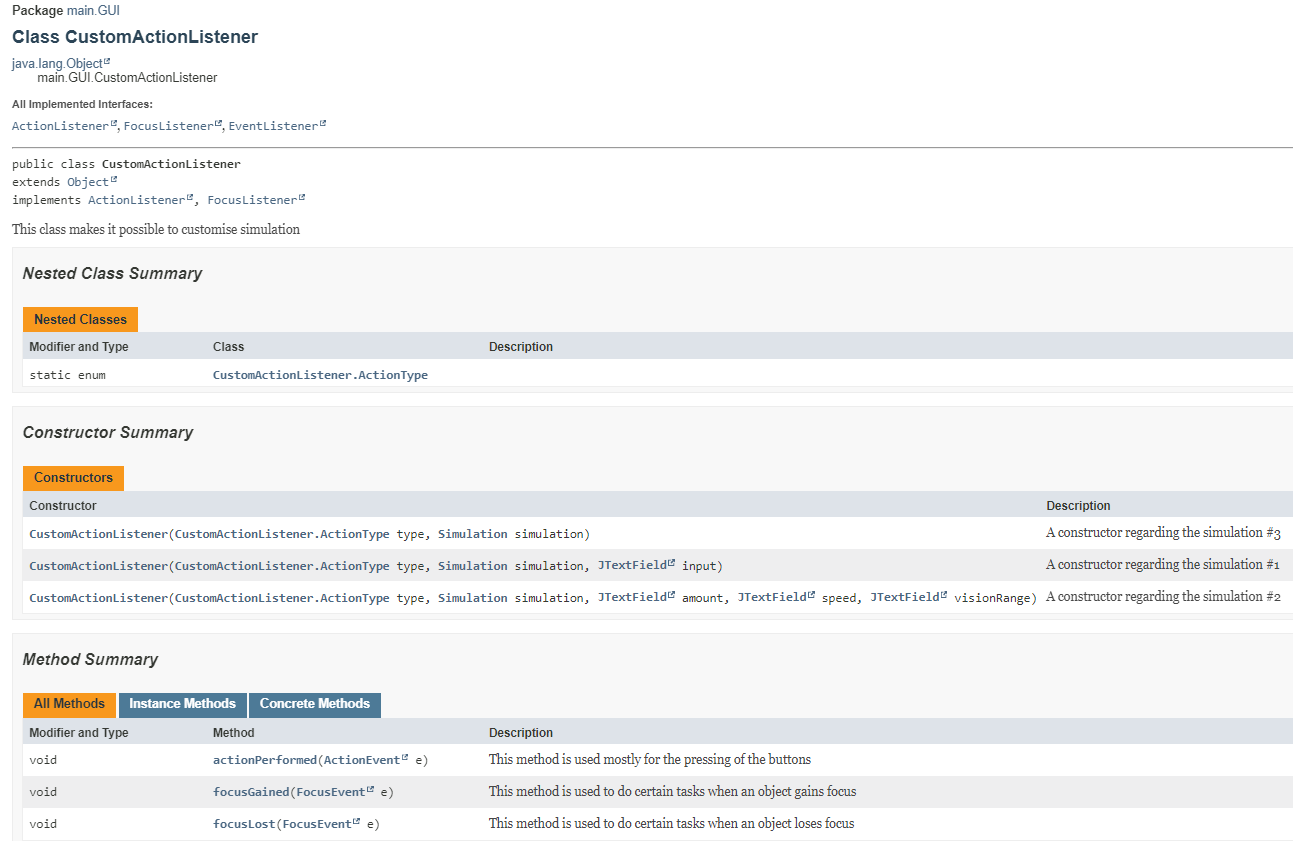
Wykonano opisy klas składających się na program, a następnie wygenerowano je. Drzew hierarchiczne wygląda następująco:



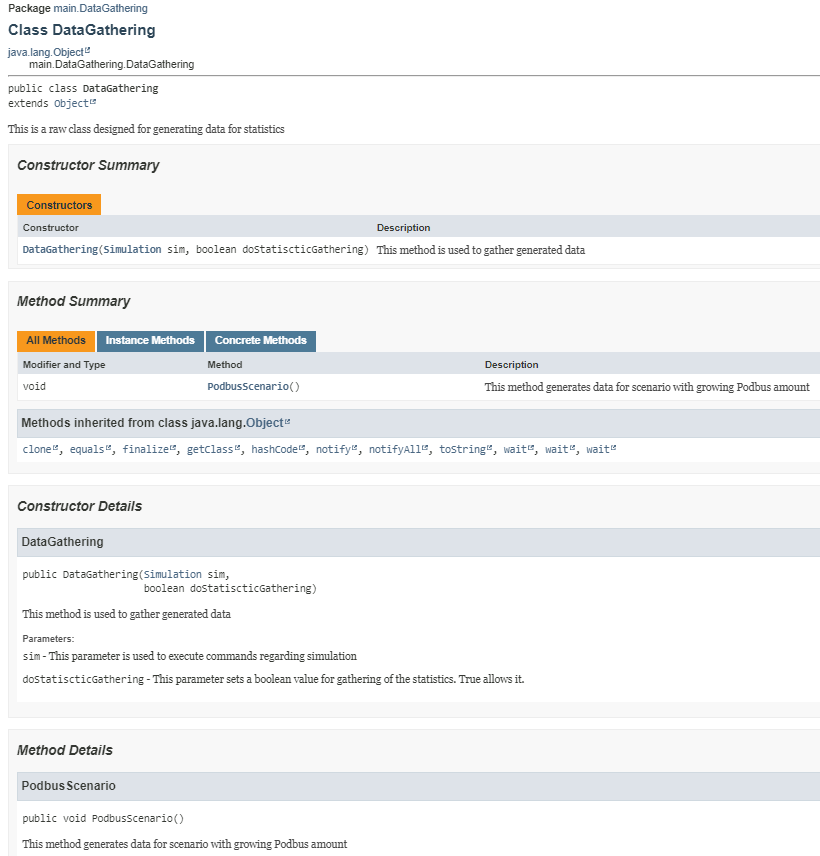
Application:



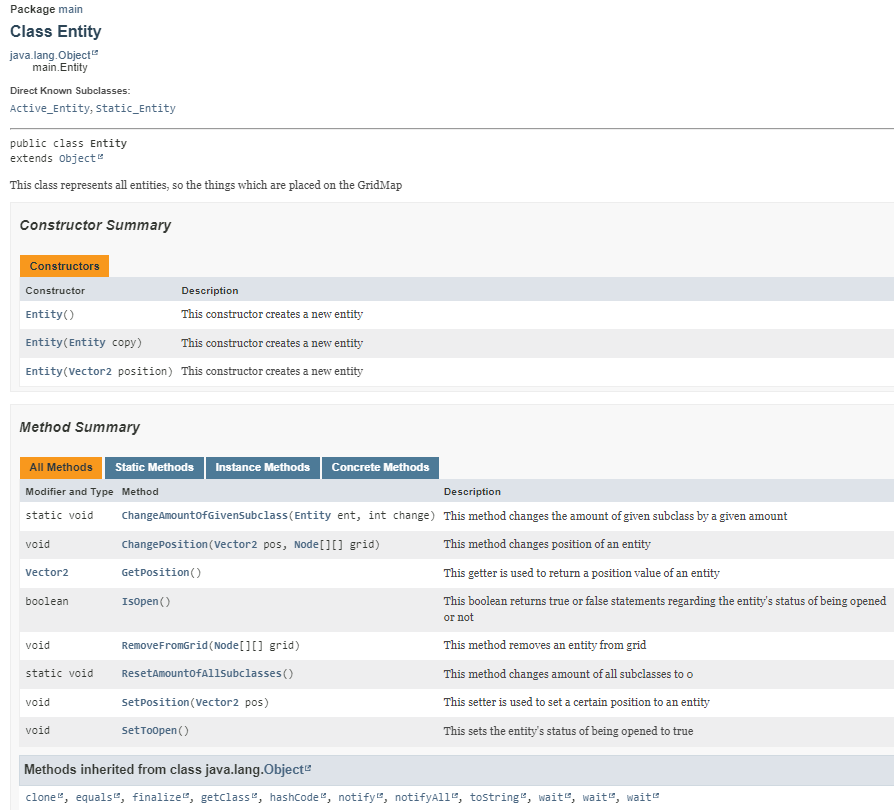
CustomActionListener:



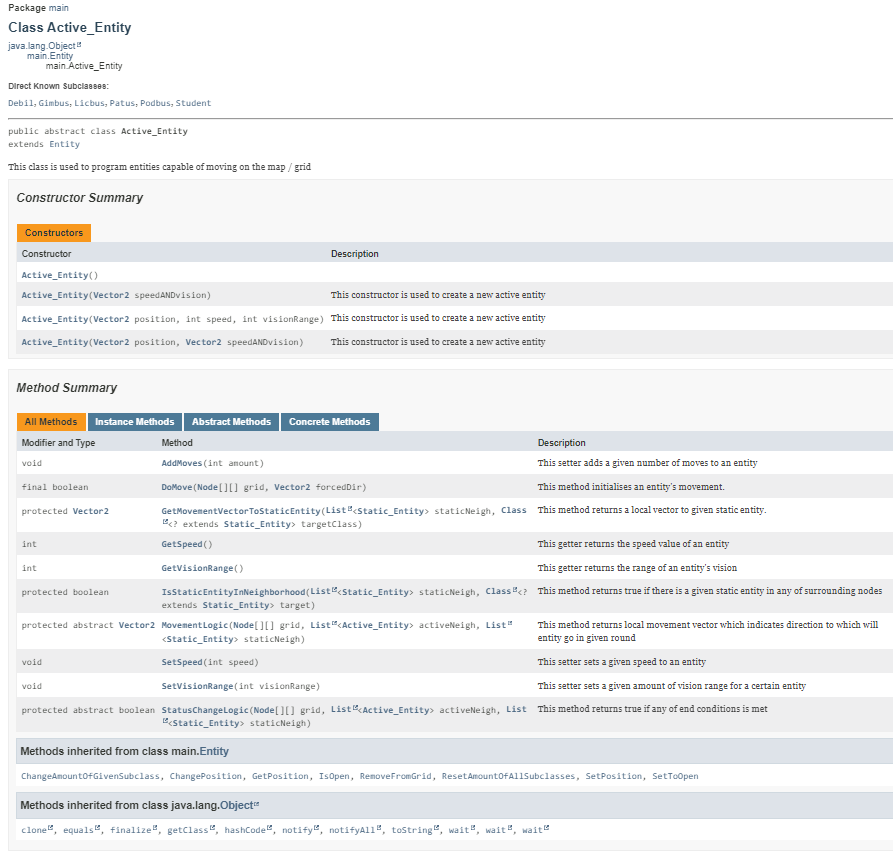
DataGathering:



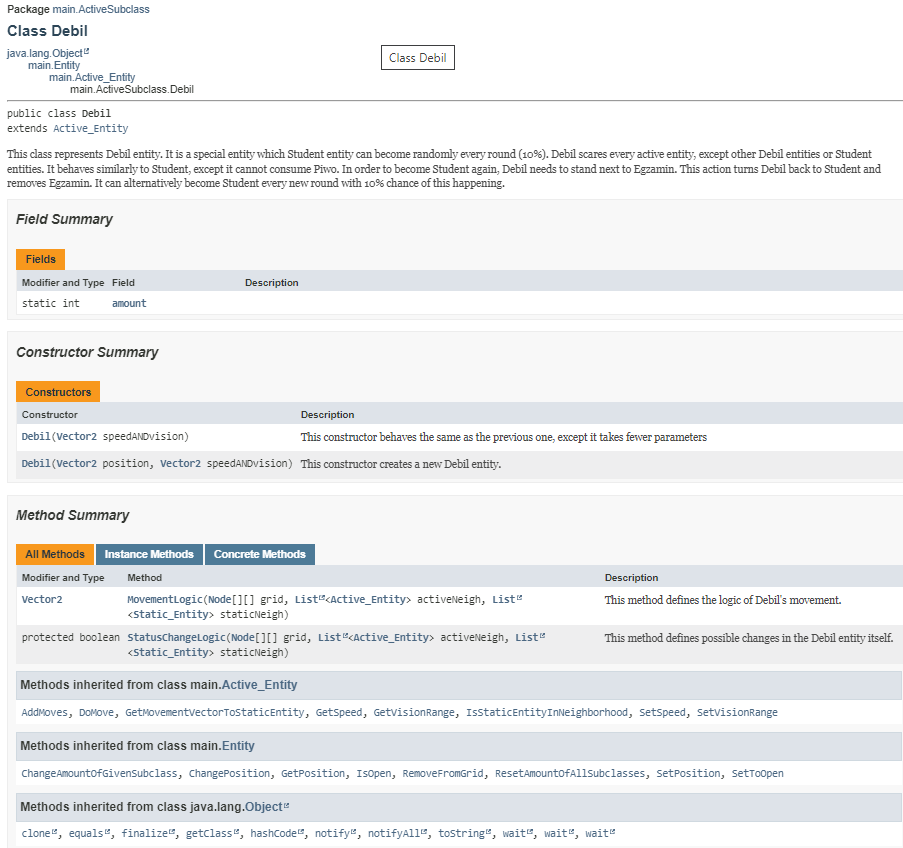
Entity:



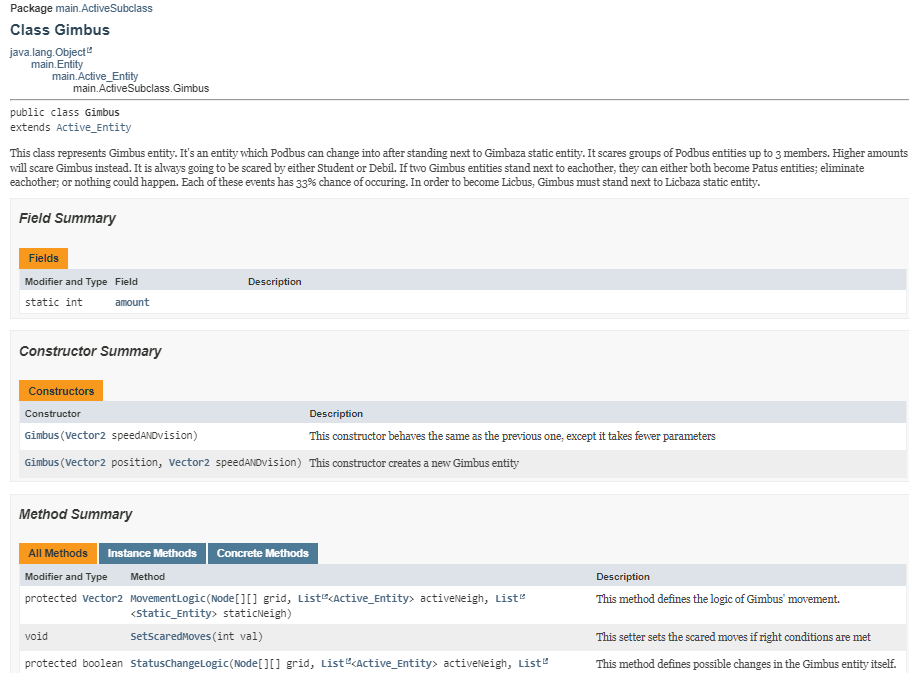
Active\_Entity:



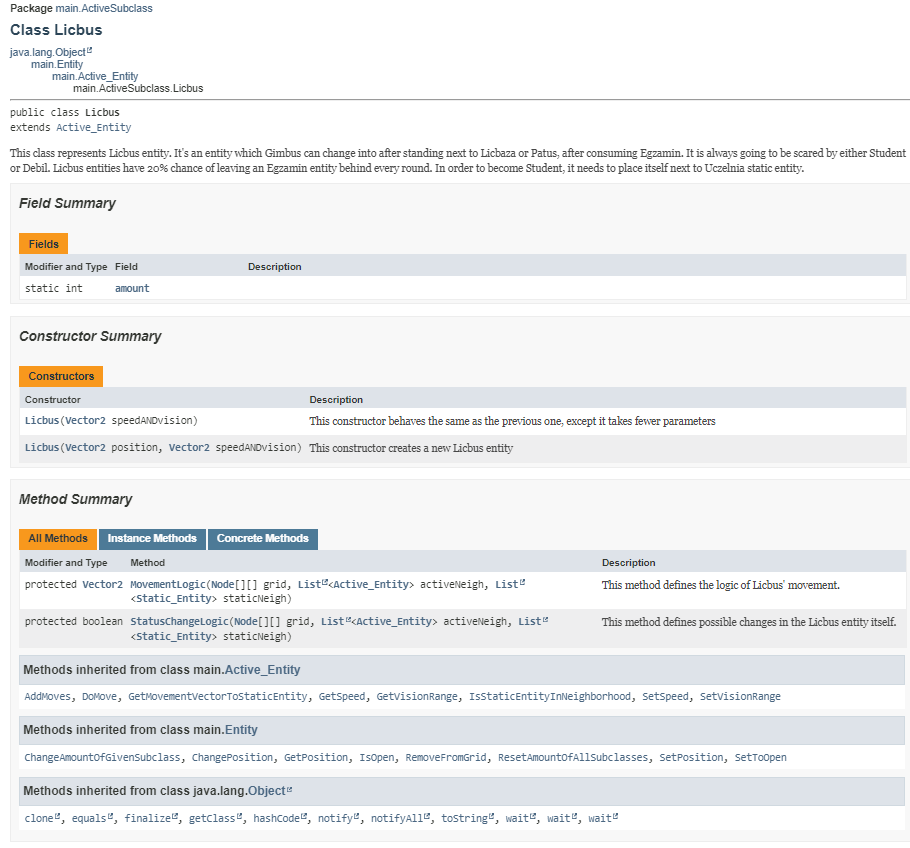
Debil:



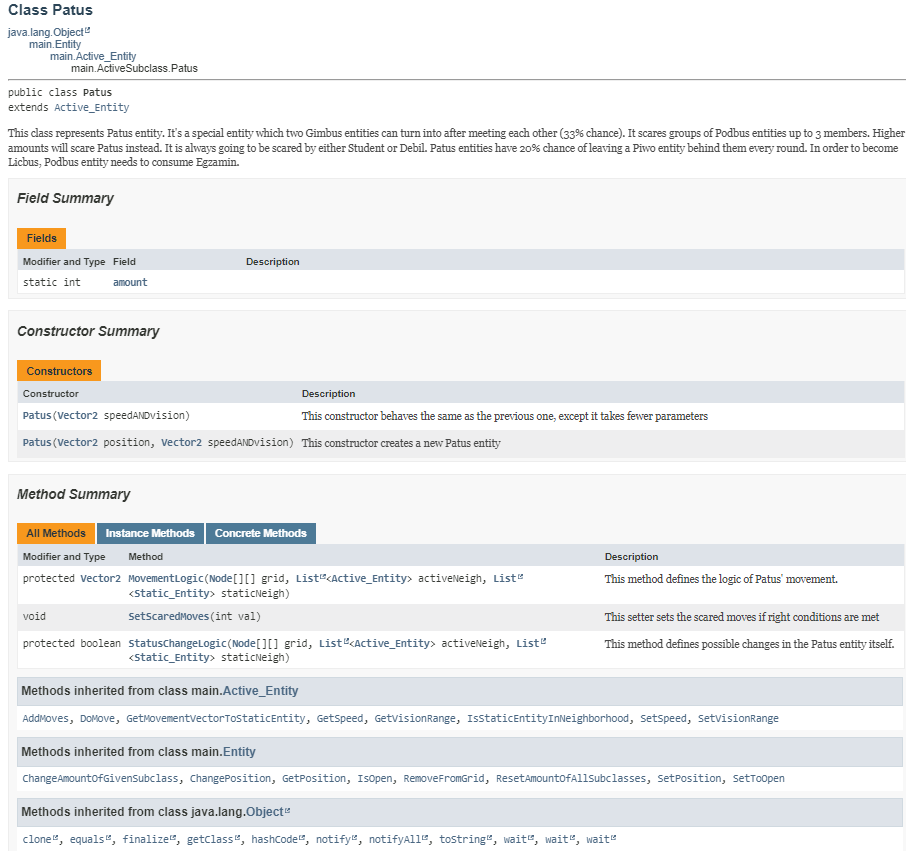
Gimbus:



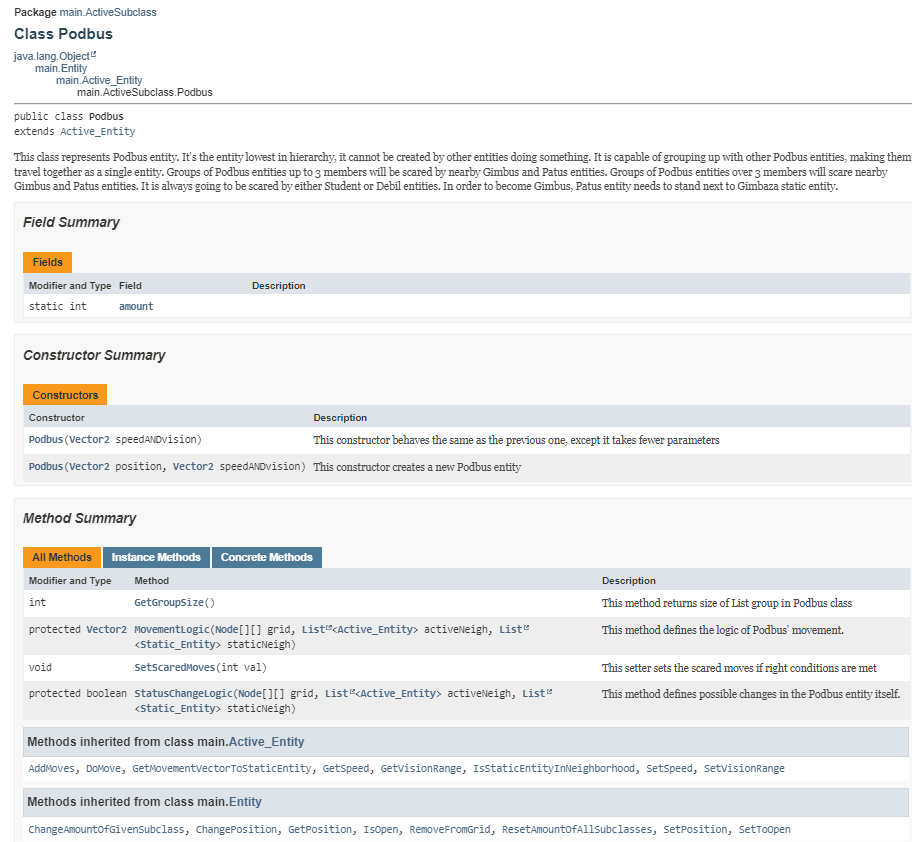
Licbus:



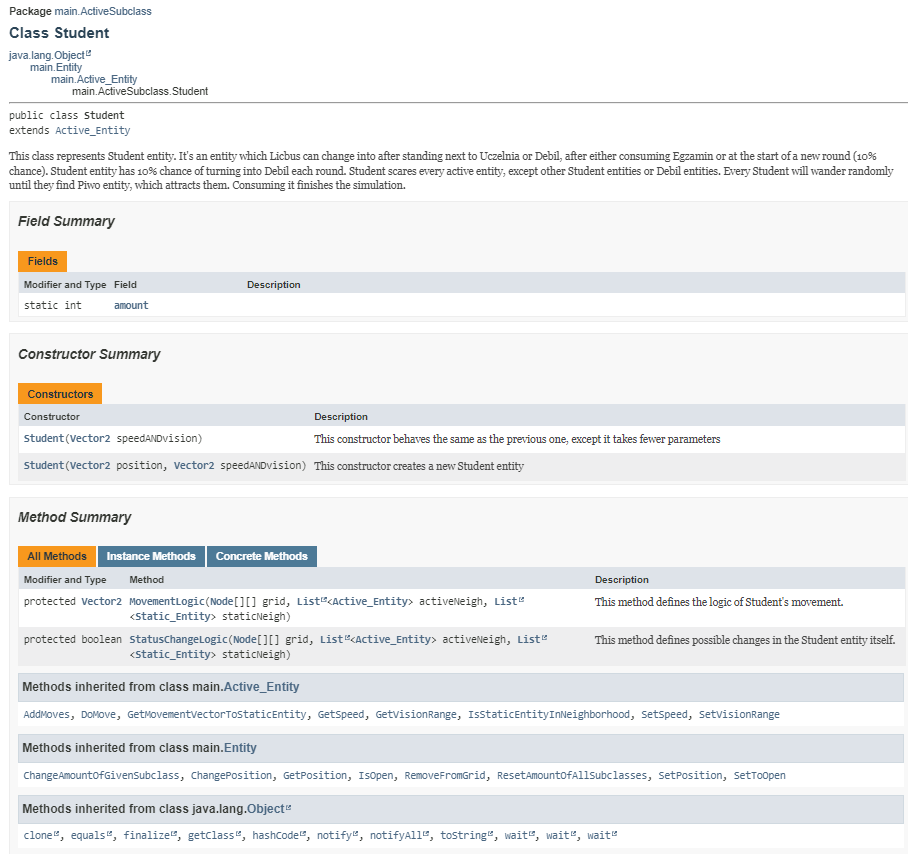
Patus:



Podbus:



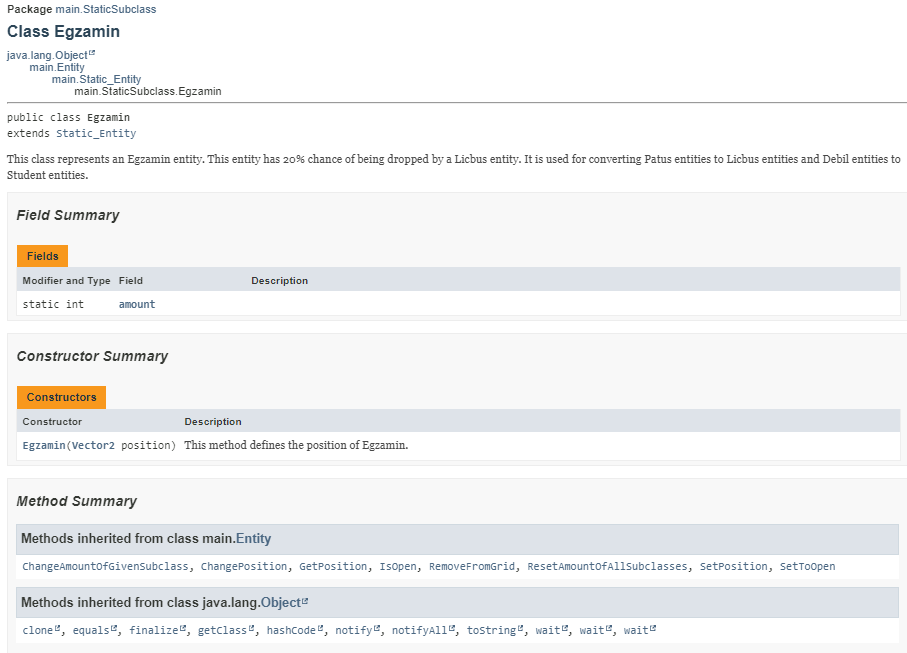
Student:



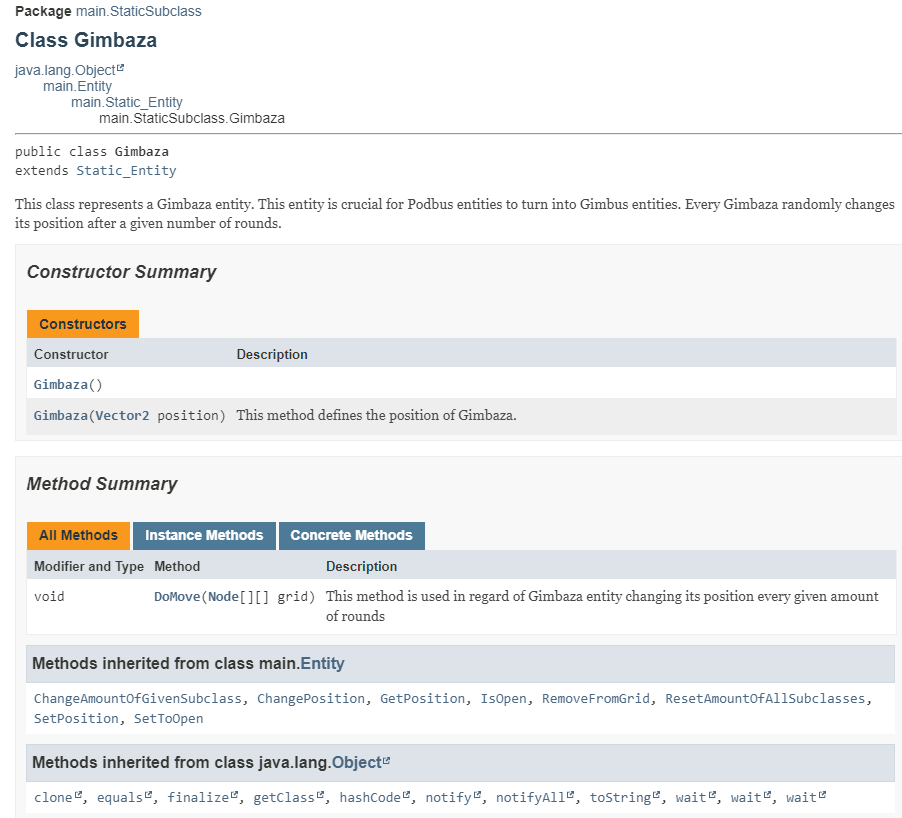
Static\_Entity:



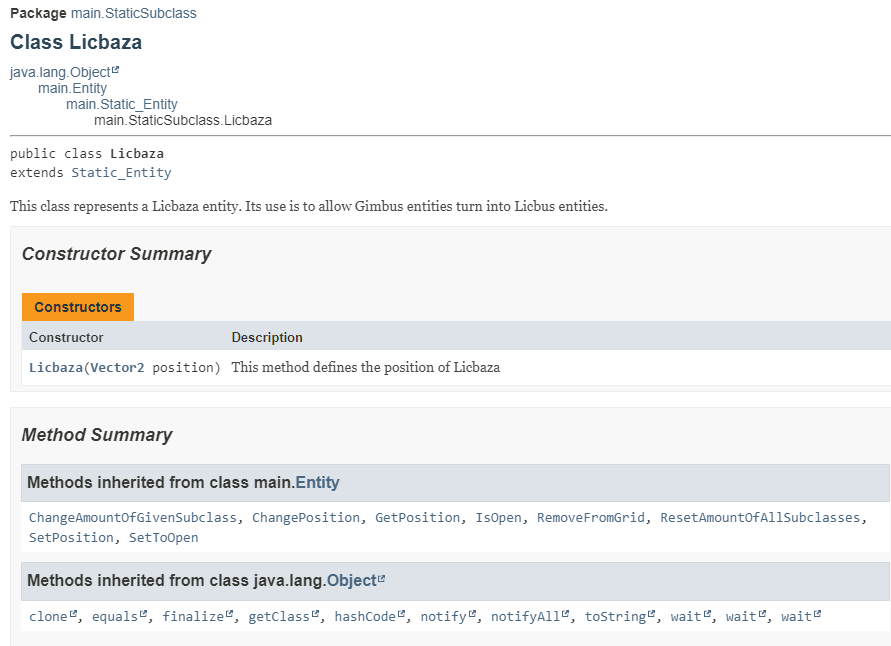
Egzamin:



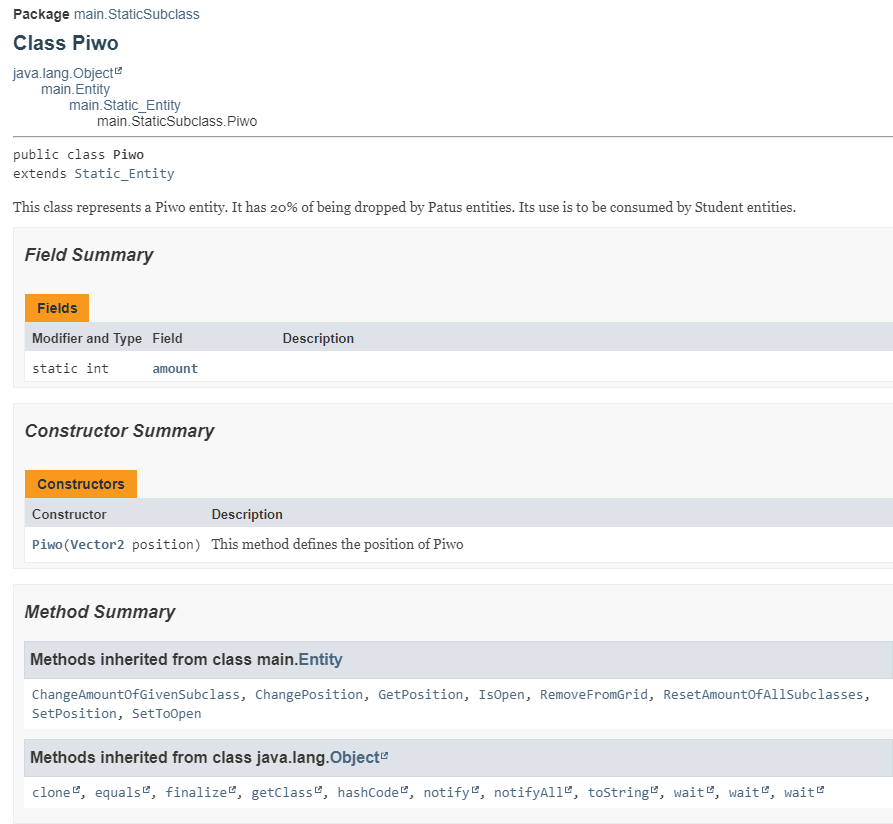
Gimbaza:



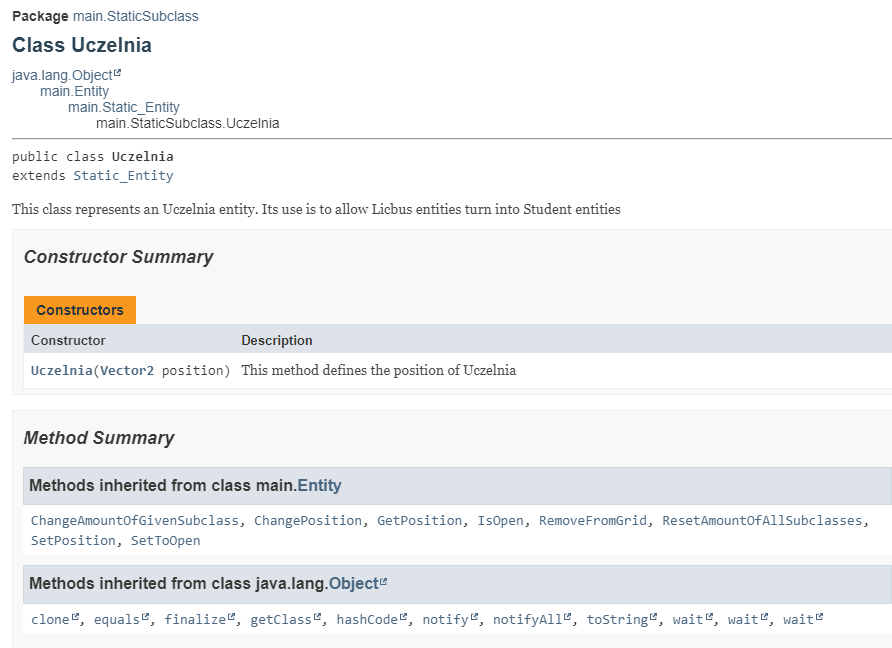
Licbaza:



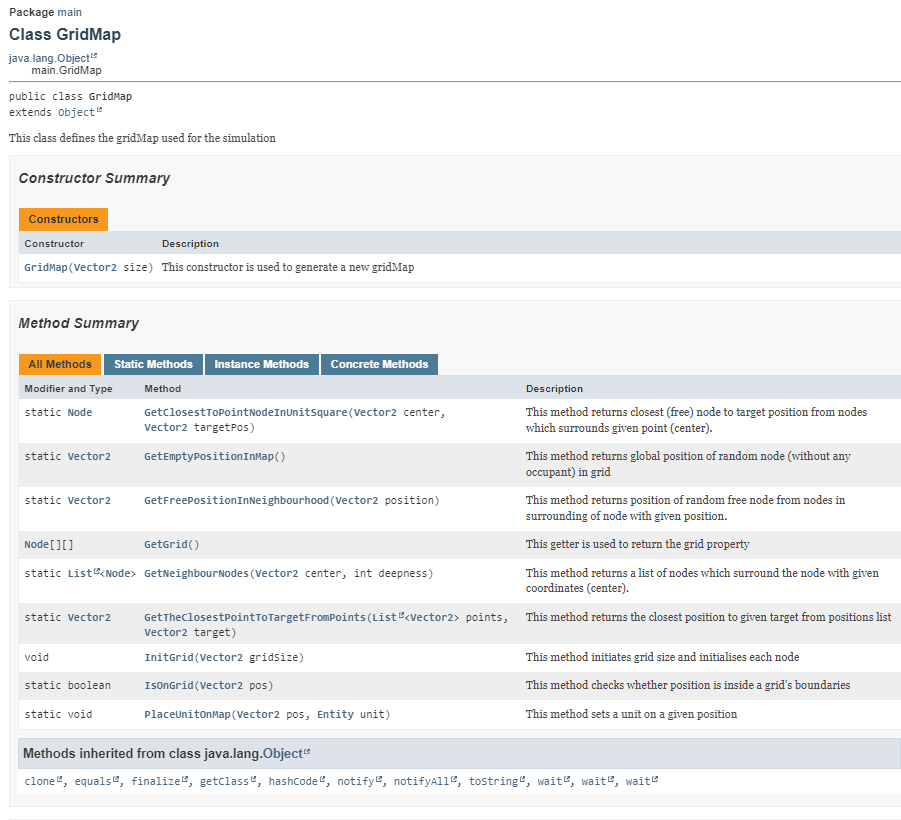
Piwo:



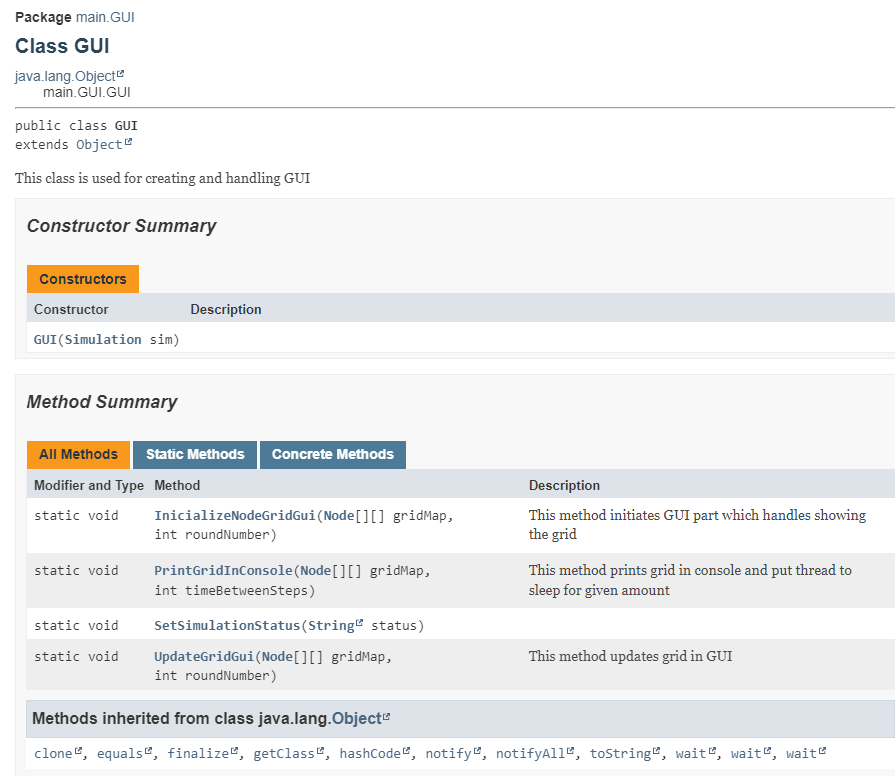
Uczelnia:



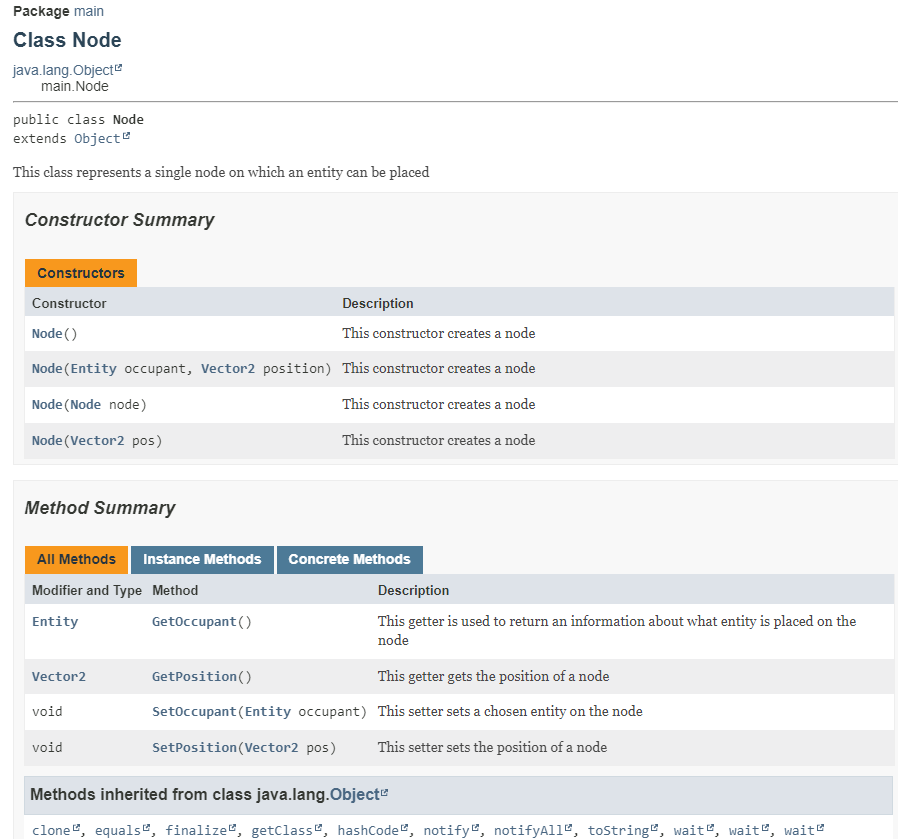
GridMap:



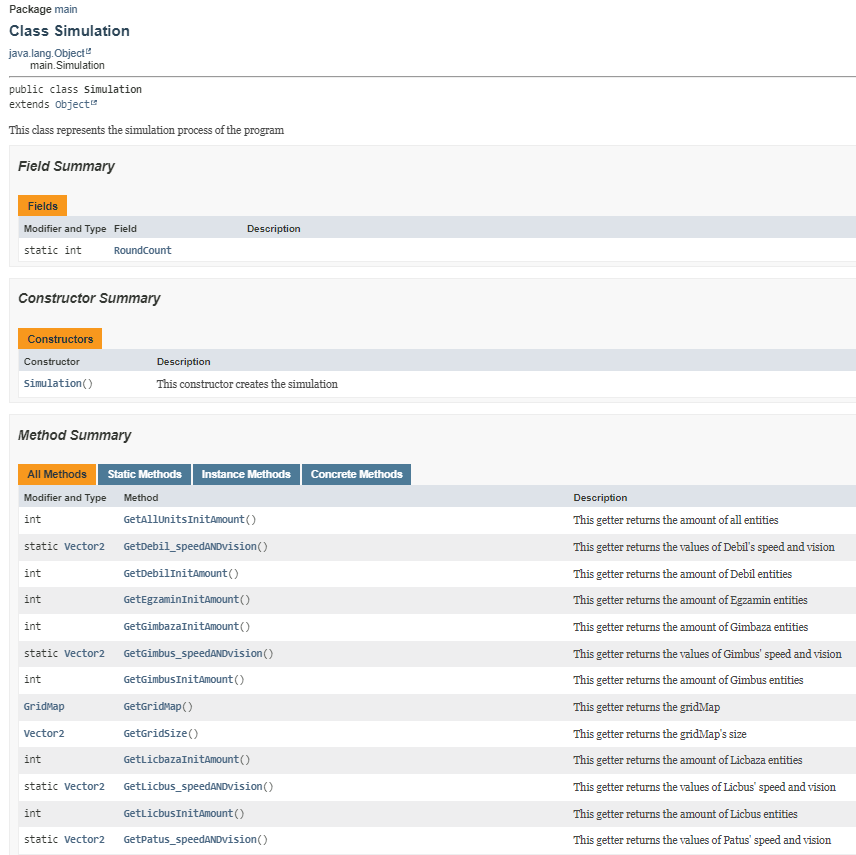
GUI:



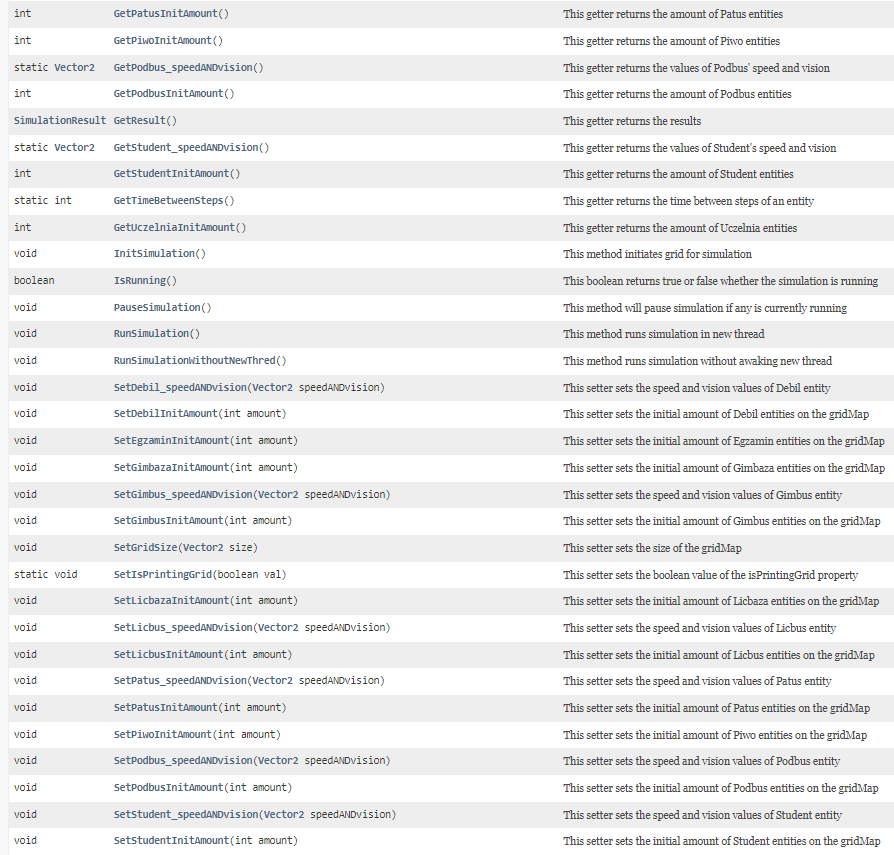
Node:



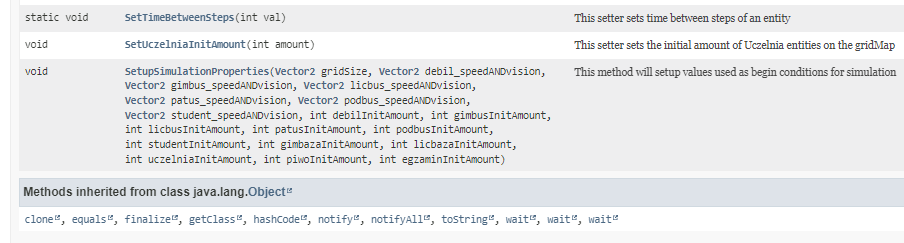
Simulation #1:



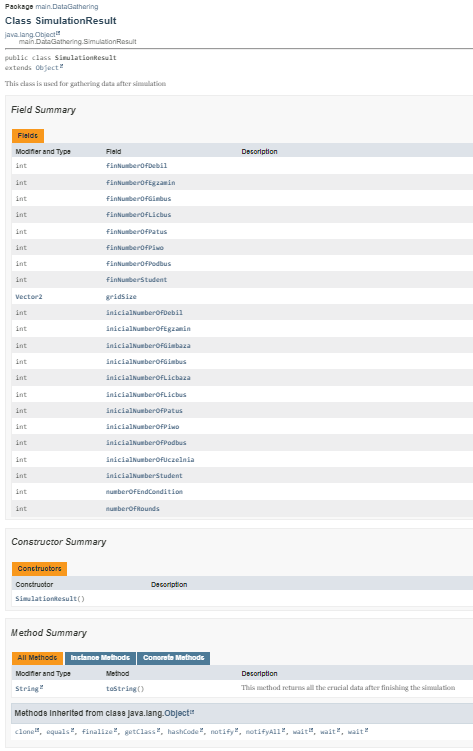
Simulation #2:



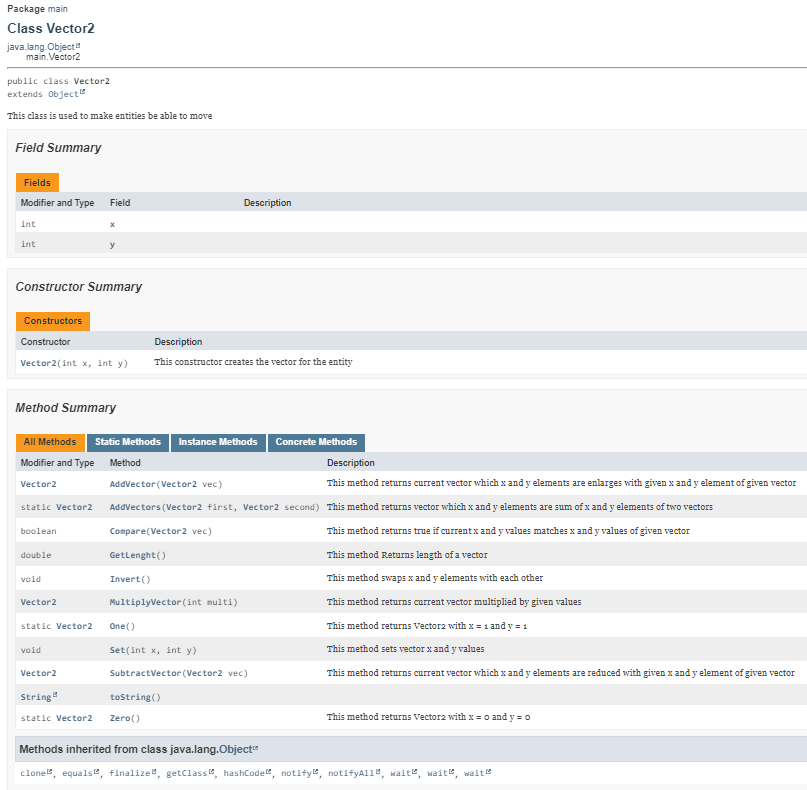
Simulation #3:



SimulationResult:

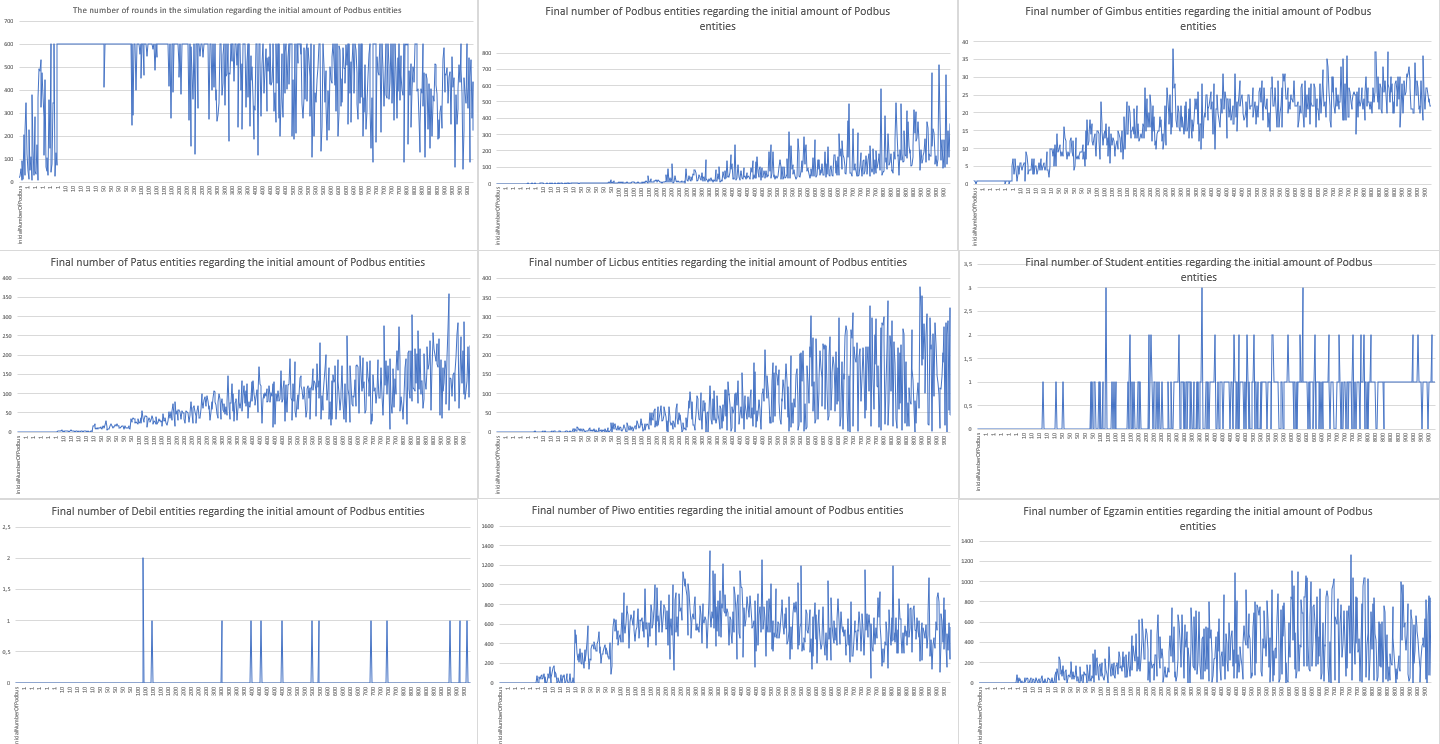


Vector2:



* Testy jednostkowe

Wykonano także testy w programie i odnotowano różne zmiany w zależności od zmiany wartości początkowej pewnego parametru. Ustalono, że byłaby to liczba Podbusów. Wyniki testów jednostkowych, przedstawiające zależności końcowych ilości Rund oraz Entitów, odnośnie początkowej liczby Podbusów, wyglądają następująco:



Widać bardzo duże rozrzuty wartości końcowych tych parametrów, jednak przy większości przypadków da się zauważyć pewne zależności:

1. Entity ruchome końcowe (Student i Debil) występują wręcz w przytłaczająco niskich ilościach w porównaniu do pozostałych Entitów.
2. Liczba rund na początku rośnie, a po przekroczeniu około 50 Podbusów, zaczyna stopniowo i powoli, maleć
3. Patusy, Licbusy i Egzaminy zdają się mieć coraz większy rozstrzał wyników wraz z postępem ilości początkowej Podbusów. Warto wspomnieć, że te jednostki mają pewną zależność. Licbusy upuszczają Egzaminy, które przyciągają Patusy. Być może wpływa to jakoś na taki rozrzut wyników wraz z postępującą ilością Podbusów.
4. Podczas, gdy liczba bytów ruchomych zdaje się rosnąć wraz z ilością Podbusów, po czym ustabilnia się, to liczba bytów nieruchomych (Piwo i Egzamin) zdaje się rosnąć do pewnej ilości Podbusów (około 300) po czym zaczyna stopniowo maleć.

* Zdjęcia aplikacji

