

Ćwiczenie 1

W tym zadaniu należy podzielić się na 3-osobowe grupy i odpowiednio rozdzielić pracę przy wykonywaniu poniższych zadań. Należy też odpowiednio rozdzielić pracę na gałęzie i commity, trzymając się przy tym dobrych praktyk dotyczących ich nazewnictwa. Na początek jedna osoba z grupy powinna sforkować projekt `starcats` i dać do niego dostęp pozostałym osobom z grupy.

Poniższe zadania rozdzielcie między siebie tak, by uniknąć merge conflictów, oraz by praca mogła odbywać się równolegle w jak największym stopniu. Przy ich wykonywaniu należy trzymać się dobrych praktyk.

1. Zaimplementuj głaskanie kota przez człowieka. Głaskanie zaczyna się losowaniem liczby całkowitej z zakresu $[0, 1]$. 0 oznacza, że pogłaskanie zakończy się atakiem na człowieka, 1 oznacza, że kot zamruczy.
2. Zaimplementuj pojedynek kotów (1 na 1) na arenie.
3. Dodaj funkcjonalność, w której jeśli człowiek pojawia się na arenie, to po bijatyce głaszcze kota z najniższym wynikiem (każdego po kolei w przypadku remisu).
4. Dodaj nowy rodzaj kota.
5. Zadbaj, by arena działała jak skrypt. Powinien on udostępniać 2 tryby: pojedynek i bijatyka. Bijatyka umożliwia podanie liczby kotów rudych i czarno-białych, które wezmą w niej udział. Pojedynek pozwala podać rodzaje i imiona kotów, które go stoczą.
6. Uzupełnij wszystkie brakujące docstringi.
7. Napisz README, opisujące co robi moduł i jak użyć skryptu.
8. Zaimplementuj utopię. Utopia jest tworzona z kotów i ludzi. Każdy człowiek w utopii raz na 10 sekund podejmuje próbę pogłaskania losowego kota. Koty kotują.
9. Utopia powinna działać jako skrypt, który przy uruchomieniu tworzy utopię z losową liczbą ludzi i losową liczbą każdego rodzaju kota. Jako argumenty można przyjąć granice rozkładu, z którego losujemy.

Po zakończeniu pracy nad `starcats` otagujcie gotową wersję jako 0.5.0. Tag należy stworzyć lokalnie i pushować na Gitlaba.

Ćwiczenie 1.5

Odtwórz na podstawie sfinalizowanego kodu z powyższego zadania schemat UML opisujący moduł `starcats`

Ćwiczenie 2

Napisz testy sprawdzające poprawność działania wszystkich funkcjonalności w module `starcats`.

Ćwiczenie 3

Napisz skrypt CI/CD na Gitlaba, który będzie podzielony na stadia:

1. `build` (czy pakiet się w ogóle daje zbudować).
2. `test` (testy jednostkowe, weryfikacja pep-8 i typowania).
3. `deploy` (uruchamia się tylko, gdy jest pushowany tag. Wypisuje w konsoli wybrany łamacz językowy).