

Technologie programowania - 2022

Lista 4

Warcaby

Celem jest zaprojektowanie i implementacja systemu do prowadzenia gry w warcaby.

System należy tworzyć **w parach wykorzystując repozytorium (np. Github)**. Historia zmian w repozytorium będzie podstawą do oceny wkładu członków zespołu.

Zasady gry

Zasady są powszechnie znane. Można je zobaczyć: [tutaj](#). Należy zaimplementować **trzy różne warianty gry** z tabeli **Warianty warcabów** znajdujące się w sekcji **Różne odmiany gry**.

Wymagania dotyczące systemu

1. System powinien działać w oparciu o architekturę klient-serwer. W razie wątpliwości warto przyrzeć się przykładom z [tej strony](#).
2. Gracz za pomocą aplikacji klienckiej powinien móc połączyć się z serwerem, dołączyć do gry i przeprowadzić rozgrywkę.
3. Serwer powinien weryfikować poprawność ruchów i pośredniczyć w komunikacji.
4. Należy odseparować warstwę modelu od warstwy widoku. Można w tym celu użyć wzorca MVC.

Dodatkowe wymagania

1. Postaraj się tworzyć system starannie i w przemyślany sposób. Wykorzystaj UML by zaproponować i rozwijać projekt systemu. Systemy chaotyczne zaprojektowane i pisane na ostatnią chwilę będą nisko ocenione. Bierz również pod uwagę, że to co stworzysz będzie miało wpływ na kolejną listę.
2. Zaimplementuj testy jednostkowe.
3. Stosuj podejście iteracyjne i przyrostowe.
4. Postaraj się wykorzystać poznane narzędzia i wzorce. Im więcej ich zastosujesz, tym wyżej będzie ocenione Twoje rozwiązanie. Ich użycie powinno być jednak zawsze uzasadnione.

Punktacja

Za implementację całej funkcjonalności każdy student może otrzymać maksymalnie **100 punktów**. Poprawność kodu powinna być na bieżąco weryfikowana testami jednostkowymi napisanymi przy użyciu JUnit (**40 punktów**). Stopień pokrycia testami może zostać zweryfikowany przez prowadzącego. Na bieżąco wysyłaj dobrze opisane zmiany do repozytorium (**20 punktów**). Przy oddawaniu projektu powinieneś opisać jego strukturę i działanie istotnych elementów – stwórz w tym celu diagram klas i inne diagramy UML, które mogą być pomocne (**20 punktów**). Stwórz dokumentację Javadoc (**10 punktów**). Łącznie każdy student może otrzymać **190 punktów**.

Sposób zaliczenia

Pokazanie rozwiązania prowadzącemu oraz wrzucenie na ePortal linku do repozytorium. Projekt będzie podzielony na dwa sprinty. Pierwszy powinien być zakończony do Świąt. Zakończenie drugiego planowane jest na pierwsze zajęcia po Nowym Roku.