

Propozycje programów zaliczeniowych

Wymagania ogólne dotyczące projektów zaliczeniowych:

- Implementacja projektu powinna zostać poprzedzona zamodelowaniem problemu za pomocą diagramu klas UML
- W projekcie musi znaleźć się co najmniej jedna klasa generyczna
- Każda klasa powinna zostać rozdzielona na dwa pliki: **.cpp* oraz **h* (lub **.hpp*). W pliku
- Wykorzystanie technik polimorficznych.
- Wykorzystanie wyjątków.
- Zastosowanie techniki enkapsulacji (ukrycie tych elementów, dla których nie ma powodu, żeby były publiczne).
- Zapisywanie i odczytywanie danych z pliku (np. lista najlepszych wyników graczy z sortowaniem, log gry (pełna historia – np. lista wszystkich ruchów w przypadku gry, lista zmian dokonanych przez użytkownika w przypadku arkusza kalkulacyjnego czy edytora), wznowianie gry przerwanej przez użytkownika w poprzedniej sesji).
- Program powinien być odporny na błędy. Dodatkowo: np. ograniczenie możliwości przemieszczania kursora tylko do dozwolonych pól; posługiwanie się kursorem w przechodzeniu pomiędzy polami zamiast podawania za każdym razem współrzędnych ruchu gracza.
- Dla napisanego kodu należy wygenerować dokumentację w formacie HTML (np. za pomocą <http://www.doxygen.nl/>).

Informacje dodatkowe dotyczące projektów zaliczeniowych:

- Projekt można wykonywać w parach.
- W ramach realizacji zadań nie ma potrzeby wykorzystywania bibliotek graficznych.
- Gry powinny oferować funkcjonalność zgodną z ich opisem oraz regułami.
- **Projekt należy nadesłać mailem najpóźniej do 1.12.2019. Nadesłanie programu w terminie późniejszym skutkuje obniżeniem oceny o 0.5 pkt za każdy tydzień spóźnienia.**

Ocena 5.0

- pełna funkcjonalność (opisana powyżej)

Ocena 4.0

- brak do 2 wymagań z przedstawionej powyżej listy

Ocena 3.0

- Oprócz braku części wymagań dopuszczalne są błędy merytoryczne w działaniu programu, niepełna funkcjonalność i niepełna obsługa błędów.

1. Program gra „szachy”
2. Program gra „Warcaby”
3. Program gra „go”
4. Program gra „gomoku”
5. Program gra „domino”
6. Program gra „Reversi”
7. Program gra „Sokoban”
8. Arkusz kalkulacyjny

Powinien oferować podstawowe funkcje (suma, wartość średnia, wariancja, odchylenie standardowe), wykorzystanie standardowych operatorów (-, +, *, /), adresowanie komórek względne i bezwzględne, operacje takie jak wklej/wytnij i zapis do pliku.

9. Implementacja prostej aplikacji do zarządzania oglądanymi serialami. Aplikacja powinna pozwalać na wprowadzanie informacji o wybranych serialach i zezwalać na wybór spośród listy wprowadzonych seriali puli śledzonych. Ponadto aplikacja powinna mieć funkcjonalność porządkującą seriale, np.:
 - proponowanie najlepszego dnia na oglądanie,
 - szacowanie średniego czasu, który trzeba poświęcić tygodniowo na oglądanie śledzonych seriali,
 - sortowanie seriali według ocen (ręcznie wpisanych lub zaczerpniętych z filmwebu czy IMDb).
 - Aktualny status oglądanych seriali powinien być zapisywany do pliku.