**Autorzy:**

**Projekt PROI:**

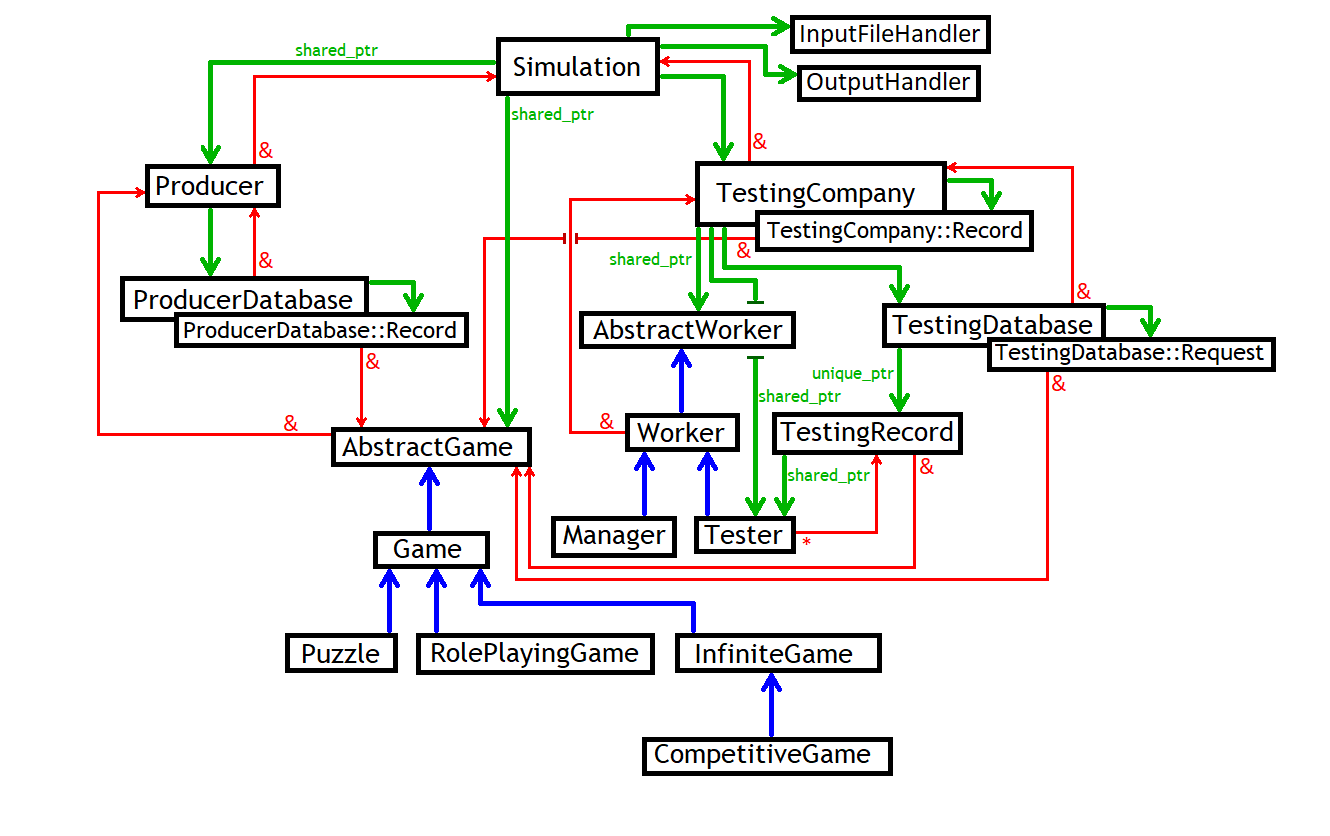
**Firma testująca gry komputerowe**

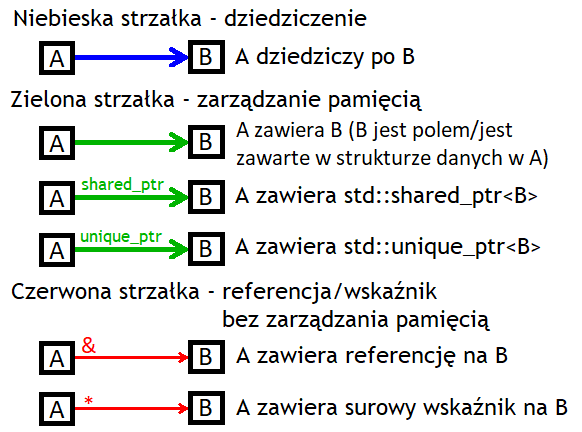
**Paweł Kochański**

**Jakub Proboszcz**

**1. Założenia projektu**

**2. Hierarchia klas**

**Schemat relacji między klasami programu:**

****Objaśnienia oznaczeń:

Na schemacie, dla lepszej czytelności, nie zostały oznaczone liczne referencje do obiektu klasy OutputHandler. Są one zawarte w klasach: Producer, ProducerDatabase, TestingCompany, Worker (i klasy potomne), TestingDatabase, TestingRecord.

**3. Opis działania symulacji**

**4. Wybrane elementy biblioteki STL**

**Elementy używane do obsługi strumieni i komunikacji z użytkownikiem:**

- **cerr** – wyjście standardowe błędu użyte do wypisania informacji o błędach w funkcji main

- **cout** – wyjście standardowe użyte do wypisywania informacji dla użytkownika

- **endl** – manipulator przejścia do nowej linii

- **ostream** – ogólny typ strumienia wyjściowego, użyty do przeładowywania operatora <<

- **stringstream** – strumień wyjściowy używany do testowania operatora <<

- **this\_thread::sleep\_for** – odczekanie określonego czasu po wypisaniu wiadomości, aby użytkownik mógł przeczytać wypisaną wiadomość

**Elementy użyte przy wykorzystywaniu zmiennych tekstowych (string):**

- **string** – podstawowy typ zmiennej składającej się z wielu znaków

- **stoi** – konwersja ciągu znaków na liczbę całkowitą

- **to\_string** – konwersja zmiennej na ciąg znaków

- **string::npos** – pokazuje że nie znaleziono danych znaków w danym ciągu znaków

**Klasy do definiowania i obsługi wyjątków:**

- **invalid\_argument** – wyjątki sygnalizujące niepoprawny argument

- **out\_of\_range** – wprowadzona liczba nie mieści się w typie danych używanym w programie

- **logic\_error** – błędy logiki programu

- **exception** - ogólny wyjątek wykorzystany przy łapaniu dowolnego wyjątku

**Kolekcje obiektów:**

- **vector** – uniwersalna kolekcja elementów w postaci ciągłej tablicy o zmiennym rozmiarze, ale stałym adresie pierwszego elementu

- **queue** – kolekcja działająca na zasadzie FIFO (first in – first out)

- **deque** – (double-ended queue, kolejka z dwoma końcami) - kolekcja pozwalająca na efektywne dodanie i usunięcie elementów z początku i końca

- **list** – kolekcja umożliwiająca efektywne dodawanie i usuwanie elementów z dowolnego jej miejsca, kosztem braku szybkiego dostępu do dowolnego jej elementu po indeksie

- **find** – znajdowanie wybranego elementu w danej kolekcji

- **find\_if** – znajdowanie elementu spełniającego podany warunek w danej kolekcji

**Generowanie liczb losowych:**

- **chrono::system\_clock::now().time\_since\_epoch().count()** – odliczenie czasu od 1 stycznia 1970 w celu wygenerowania seed’a do generatora liczb losowych

- **mt19937** – wybrany generator liczb losowych

**„Inteligentne” wskaźniki:**

- **shared\_ptr** – „inteligentny” wskaźnik do danego obiektu mogący występować w wielu kopiach. Element jest usuwany dopiero po usunięciu wszystkich wskazujących na niego shared\_ptr

- **make\_shared** – tworzy „inteligentny” wskaźnik shared\_ptr do danego obiektu

- **unique\_ptr** – „inteligentny” wskaźnik do danego obiektu mogący występować tylko w jednej kopii. Kiedy przestaje istnieć, wykonuje usunięcie obiektu, na który wskazuje

- **make\_unique** – tworzy „inteligentny” wskaźnik unique\_ptr do danego obiektu

- **move** – przesuwanie elementów (konkretnie wskaźników typu unique\_ptr) na inne miejsce w pamięci, celem uniknięcia ich kopiowania

**Zarządzanie plikami:**

- **ofstream** – strumień pliku wyjściowego

- **ifstream** – strumień pliku wejściowego

- **getline** – zapisanie wiersza (sekwencji znaków zakończonej znakiem nowej linii) z pliku wejściowego do zmiennej w programie

**5. Zdefiniowane sytuacje wyjątkowe i ich obsługa**

W programie zostały zdefiniowane następujące sytuacje wyjątkowe (klasy wyjątków):

- **ConversionError**

- niepoprawna próba skonwertowania ciągu znaków wczytanego z pliku na dany typ

- **DuplicateGameError**

- próba dodania do bazy danych producenta gry, która już się tam znajduje

- **EmptyNameException**

- próba ustawienia nazwy obiektu na składającą się wyłącznie z białych znaków lub pustą

- **EndOfFileError**

- próba wczytania danych z pliku po dojściu do jego końca

- **FileError**

- niepowodzenie otwarcia pliku (np. plik wejściowy nie istnieje)

- **GameAlreadyTestedError**

- próba rozpoczęcia testowania już przetestowanej gry

- **GameNotPresentError**

- próba zmiany gry w bazie danych producenta, kiedy ona się tam nie znajduje

- **GameNotRequestedError**

- próba oznaczenia gry w bazie danych producenta jako przetestowanej, kiedy jej testowanie nie zostało nigdy rozpoczęte

- **IncorrectAddressException**

- próba utworzenia adresu z kodem pocztowym w formacie innym niż dd-ddd  
(d - cyfra)

- **InvalidFileSize**

- próba ustawienia rozmiaru plików gry na 0 KB

- **InvalidFullLength**

- próba ustawienia pełnej długości (czasu potrzebnego na pełne doświadczenie) gry fabularnej na krótszą od bazowej długości tej gry

- **InvalidId**

- próba ustawienia identyfikatora obiektu na wartość niedozwoloną dla tej klasy obiektów

- **InvalidLength**

- próba ustawienia bazowej długości gry na 0 minut

- **InvalidPrice**

- próba ustawienia ceny na nieodpowiednią wartość (ujemna wartość, ujemna liczba groszy)

- **InvalidProducer**

- próba dodania gry wyprodukowanej przez innego producenta do bazy danych producenta

- **InvalidTestersAmount**

- próba ustawienia minimalnej liczby testerów wymaganej do przetestowania danej gry na 0, bądź próba ustawienia maksymalnej liczby testerów na mniejszą od minimalnej

- **InvalidTitle**

- próba ustawienia tytułu gry na składający się wyłącznie z białych znaków lub pusty

- **NoGamesUntestedError**

- próba pobrania gry do testów, gdy żadna nie jest dostępna.

- **TestingEndedError**

- próba modyfikacji rekordu już przetestowanej gry

- **TestingNotEndedError**

- próba odczytania końcowych parametrów nie w pełni przetestowanej gry

- **InvalidHouseNumberError**

- próba ustawienia na zero wartości numeru domu w adresie

**Obsługa sytuacji wyjątkowych**:

Wyjątki są łapane w funkcji przeprowadzającej symulację, wypisywane do pliku i na wyjście standardowe cout po czym zgłaszane jest **ShutdownException** sygnalizujące ową sytuację wyjątkową dla programu. ShutdownException łapane jest w funkcji main, co kończy tym samym symulację wypisując informację o tym na wyjście cerr.

Poza tym wyjątek **EndOfFileError** jest wykorzystywany przy tworzeniu producentów by zakończyć ich tworzenie bez potrzeby informowania programu o ich ilości.

**6. Podział obowiązków**

**Paweł Kochański:**

Klasy: Address, Worker, Manager, Tester, TestingCompany, InputFileHandler, OutputHandler, Simulation, interfejs AbstractWorker

Wyjątki związane z ww. klasami

Testy jednostkowe ww. klas

**Jakub Proboszcz:**

Klasy: Price, Game, Puzzle, CompetitiveGame, InfiniteGame, RoleplayingGame, Producer, ProducerDatabase, TestingDatabase, TestingRecord, interfejs AbstractGame

Wyjątki związane z ww. klasami

Testy jednostkowe ww. klas

**Wspólnie:**

Naprawianie zauważonych błędów i problemów w plikach drugiej osoby

Main.cpp

Sprawozdanie

Testowe pliki wejściowe

Testy integracyjne - debugowanie programu jako całości