

# DOKUMENTACJA PROJEKTU „HOTEL”

## WPROWADZENIE

Założeniem projektu jest symulacja hotelu i jego działania na przestrzeni określonej liczby dni. Symulacja obejmuje m.in. rezerwowanie pokoi przez gości, zmiana długości pobytu przez gościa, zatrudnianie i zwalnianie pracowników, itp. Symulacja opiera się na danych wejściowych pobieranych z odpowiednich plików tekstowych oraz z argumentów wywołania programu, a jej przebieg ma charakter (pseudo)losowy.

## STRUKTURA PROJEKTU

Projekt został wykonany w języku C++ zgodnie z podejściem obiektowym. Składa się on z następujących klas:

- **Hotel**

Zasadnicza klasa reprezentująca cały Hotel. Zawiera klasy reprezentujące bazy danych – pokoi i pracowników, oraz menu restauracyjne, oraz posiada informacje o budżecie (stanie konta) – budżet na start wynosi 10 000 zł

- **Bazy danych – Data\_room i Data\_employees**

Klasy posiadające listy (std::vector) przechowujące wskaźniki (std::unique\_ptr) na obiekty typu Room – od pokoi hotelowych, oraz Employee – od pracowników. Zaimplementowane w niej są metody dodające i usuwające z niej elementy oraz zwracające pewne informacje o jej elementach (czy dany pokój jest wolny w danym przedziale czasowym, ilu jest w hotelu pracowników danego typu, itp.)

- **Menu**

Klasa przechowująca kolekcję potraw i napoi – obiekty klasy Dish – składające się ze składników (Ingredient). Do menu można dodawać i usuwać potrawy i napoje, a każde z nich ma przypisaną swoją cenę

- **Interfejsy – Room\_Interface i Employee\_Interface**

Klasy abstrakcyjne zawierające czysto wirtualne metody klas odpowiednio Room i Employee, które po nich dziedziczą

- **Room**

Klasa bazowa wszystkich pokoi znajdujących się w hotelu. Pokój zawiera powierzchnię, cenę, liczbę łóżek oraz informacje, czy pokój jest podwyższonego standardu (np. ma lepsze wyposażenie) – takowy ma większą powierzchnię i cenę niż odpowiadający mu pokój niższego standardu; oraz informację czy pokój zawiera łóżko małżeńskie, czy same pojedyncze. Nadto znajduje się w nim wektor zawierający informację, w których dniach dany pokój jest zajęty. Po tej klasie dziedziczą inne klasy określające typ pokoju:

1. **Pokój jednoosobowy (OnePersonRoom) – nigdy nie ma łóżka małżeńskiego**
2. **Pokój dwuosobowy (TwoPersonRoom)**
3. **Pokój trzyosobowy (ThreePersonRoom)**
4. **Pokój czteroosobowy (FourPersonRoom)**
5. **Studio (Studio) – pokoje niższego standardu, bez łoża małżeńskiego**
6. **Apartament (Apartment) – wyższego standardu, z łóżem małżeńskim**

- **Employee**

Klasa bazowa reprezentująca wszystkich pracowników pokoju. Każdy pracownik ma imię, nazwisko, adres email, numer PESEL, godzinną stawkę oraz wektor zawierający dni, w których pracuje (grafik). W hotelu obowiązuje 3-zmianowa organizacja pracy, ale nie wszyscy pracownicy pracują na 3 zmiany. Dziedziczą po niej klasy określające typ pracownika (obok podano, w ilozmianowym systemie pracuje):

1. **Barman – 2 zmiany**
2. **Ochroniarz (Bodyguard) – 3 zmiany**
3. **Kucharz (Cook) – 2 zmiany**
4. **Pokojówka (Maid) – 3 zmiany**
5. **Manager – 1 zmiana**
6. **Recepcjonista (Receptionist) – 3 zmiany**
7. **Kelner (Waiter) – 2 zmian**

- **Guest**

Klasa reprezentująca gościa hotelowego. Gość ma imię, nazwisko, email, PESEL oraz budżet (określoną sumę pieniędzy na koncie). Może on rezerwować pokój zgodnie z określonymi przez siebie preferencjami, zamówić taksówkę, sprzątanie pokoju oraz budzenie. Nadto może zamawiać jedzenie z restauracji.

- **Date**

Klasa pomocnicza reprezentująca pojedynczy dzień w kalendarzu. Ma ona przeciążone operatory pozwalające na dodawanie i odejmowanie dni od daty,

obliczanie różnicy między dwoma datami (wszystko z uwzględnieniem lat przestępnych). Jest ona używana w grafikach pracowników oraz w rezerwacjach pokoi.

- **Simulation**

Klasa odpowiedzialna za przebieg symulacji – pobieranie danych z plików tekstowych, losowanie czynności w trakcie jej trwania, wyświetlanie informacji na ekranie oraz zapisywanie ich do pliku tekstowego

## **INSTRUKCJA UŻYTKOWNIKA**

Jak już wyżej wspomniano, do poprawnego działania symulacji niezbędne są pliki tekstowe zawierające – potencjalnych gości (pulę, z której będą losowani), pracowników, zestaw pokoi oraz menu. Struktura tych plików powinna być następująca:

- **Plik z gośćmi**

Każdy gość w osobnej linii, dane, kolejno – imię, nazwisko, email, PESEL, budżet (posiadane pieniądze na konie, wyrażone liczbą całkowitą) – wszystkie oddzielone pojedynczą spacją

Przykładowa zawartość:

```
Jan Mazur jan.mazur@wp.pl 02232123357 20000000  
Krystof Tyniec kt@interia.pl 09923454312 300000  
Robert Gawlinski rg@onet.pl 93445812031 303000  
Marek Gawlinski rg@onet.pl 93445812131 303000  
Janusz Gawlinski rg@onet.pl 93445852031 303000  
Dawid Gawlinski rg@onet.pl 73445812031 303000  
Szymon Gawlinski rg@onet.pl 93945812031 303000  
Piotr Gawlinski rg@onet.pl 93445844441 303000
```

- **Plik z pracownikami**

Każdy pracownik w osobnej linii, dane, kolejno – imię, nazwisko, email, PESEL, typ (jedno ze słów – **manager, cook, bodyguard, barman, maid, recepcjonist, waiter**), stawka godzinowa wyrażona liczbą (może być zmiennoprzecinkowa) – wszystkie oddzielone pojedynczą spacją. Aby hotel efektywnie działał i dobierał pracownikom grafik pracy zaleca się dodanie kilku (przynajmniej sześciu) pracowników każdego typu

Przykładowa zawartość:

Regina Celi orapronobis@deus.pl 00301123888 maid 22.5  
Anastazja Nowak anasta@wp.pl 92324588729 recepcjonist 20  
Karolina Trok mala@trello.pl 03250523625 recepcjonist 19.7  
Leyla Lema honey@onet.eu 98071823923 recepcjonist 21.6  
Marek Wojcik wojcikiewicz@gmail.com 00301130333 recepcjonist 21  
Kuba Nowaczyski razbery@wp.pl 98324588729 bodyguard 20

- **Plik z pokojami**

Każdy pokój w osobnej linii, dane, kolejno – numer pokoju (liczba całkowita dodatnia), znak określający typ pokoju (1 – jednoosobowy, 2 – dwuosobowy, 3 – trzyosobowy, 4 – czteroosobowy, s – studio, a – apartament), dwie wartości (true lub false) określające kolejno – czy pokój jest podwyższonego standardu oraz czy zawiera łożę małżeńskie – w przypadku pokoi dwu-, trzy- i czteroosobowych, zaś dla pokoi jednoosobowych wystarczy tylko jedna taka wartość – łóżka małżeńskiego pokój jednoosobowy nie posiada. Pokoje typu studio i apartament nie wymagają tych dwóch dodatkowych wartości, gdyż mają je przypisane automatycznie

Przykładowa zawartość:

1 1 false false

2 1 false

3 3 false true

4 4 true true

5 s

6 s

7 s

8 s

9 s

10 a

11 a

12 a

13 a

14 a

- **Plik z menu**

Każda potrawa w osobnej linii, dane, kolejno – słowo „dish” lub „drink” określające czy dana rzecz jest potrawą czy drinkiem, nazwa, cena, koszt przygotowania, czas przygotowania, składniki oraz ich ilość w potrawie/drinku – oddzielone pojedynczymi spacjami

Przykładowa zawartość:

```
dish Pizza_Margherita 22 10 120 sause 15 cheese 50 dough 100
dish Pizza_Capriciosa 25 15 120 sause 15 cheese 50 mushrooms 30 ham 25 dough 100
dish Pizza_Quatro_Formaggi 30 15 120 sause 15 gouda 20 feta 15 mozzarella 20 blue 10
drink Kamikadze 10 3 1 vodka 10 liquer 15
```

### **Uruchomienie i przebieg**

Po upewnieniu się, że w katalogu projektowym znajdują się wszystkie pliki wejściowe należy zdefiniować odpowiednie argumenty wywołania programu – kolejno, data startowa (format DD.MM.RRRR), liczba dni, przez które będzie się toczyć symulacja, oraz nazwy plików tekstowych kolejno – z zestawem pokoi, zestawem pracowników, menu, gośćmi, oraz nazwa pliku do którego zostanie zapisany przebieg symulacji (UWAGA! – jeżeli w folderze istnieje plik o tej samej nazwie, zostanie nadpisany).

W czasie trwania programu na ekranie będą stopniowo wyświetlać się krótkie komunikaty, np. Guest with PESEL 93445844431 order\_waking\_up at 23. Ten sam zestaw komunikatów zostanie zapisany do pliku. Na samym końcu symulacji pojawi się informacja podsumowująca – ile pracowników jest zatrudnionych w hotelu, ilu gości się przez niego przewinęło oraz jaki jest stan konta na końcu symulacji.

## Przykładowy przebieg – dla trzech dni

```
Guest with PESEL 93445844471 order_waking_up at 21
It is not possible to book a room for Piotr Gaw
It is not possible to book a room for Piotr Gawl
It is not possible to book a room for Piotr Gawlinski
Guest with PESEL 93445844471 order_dish Pizza_Capriciosa
Guest with PESEL 93445844471 order_dish Pizza_Capriciosa
Janusz Gawlinski booked the room nr 35
Guest with PESEL 93445852031 order_waking_up at 21
CHANGE! Employees on change:
Janek Baranek maid
Karolina Trok receptionist
Malgosia Lipa bodyguard

Guest with PESEL 93445844471 order_dish Pizza_Capriciosa
Guest with PESEL 93445844471 order_dish Pizza_Capriciosa
Piotr Gawli booked the room nr 127
Guest with PESEL 93445852031 order_dish Pizza_Capriciosa
Guest with PESEL 93445844431 order_waking_up at 21
Guest with PESEL 93445844431 order_waking_up at 22
Next day: 5.1.2020
CHANGE! Employees on change:
Agnieszka Mazur manager
Robert Mazurek cook
Magda Beza maid
Leyla Lema receptionist
Kuba Nowaczyski bodyguard
Zbigniew Stonoga barman
Ozyrys K barman
Ola Siwiec waiter
Ela Siwiec waiter

Summary:
Hotel have 34 employees
Hotel had 3 guests
Hotel's final budget 16375.0

Process returned 0 (0x0)   execution time : 0.086 s
Press any key to continue.
```

## PODZIAŁ RÓL W ZESPOLE

<b>Zadanie</b>	<b>Osoba odpowiedzialna</b>
Stworzenie interfejsu i klas bazowych oraz pochodnych dla pracowników hotelu	<b>Dominika Wyszyńska</b>
Stworzenie interfejsu i klas bazowych oraz pochodnych dla pokoi hotelowych	<b>Szymon Spaczyński</b>
Menu restauracyjne, klasy Menu, Dish i Ingredient, klasa pomocnicza Date	<b>Szymon Spaczyński</b>
Klasa główna Hotel	<b>Dominika Wyszyńska (plik nagłówkowy, metody do obsługi pracowników)</b> <b>Szymon Spaczyński (implementacja metod do pokoi oraz obsługi gości)</b>
Klasa Simulation, zaplanowanie przebiegu symulacji	<b>Dominika Wyszyńska</b>
Klasa Guest	<b>Dominika Wyszyńska</b>
Obsługa plików i argumentów wejściowych	<b>Szymon Spaczyński</b>
Testy jednostkowe	<b>Dominika Wyszyńska (Hotel, Employee, Data_employee)</b> <b>Szymon Spaczyński (Room, Data_room, Menu, Date)</b>
Logiczne zaplanowanie działania projektu	<b>Dominika Wyszyńska</b>
Naprawa błędów, poprawianie jakości kodu, refaktoryzacja	<b>Szymon Spaczyński</b>
Sporządzenie dokumentacji projektowej	<b>Szymon Spaczyński</b>