

# Student Simulator 2021

*Made by:*

*Szymon Paluszewski 58193 - Program, Klasa Random Event, Klasa Sklep*

*Bartosz Suska 58203- Program, schemat UML, Klasa Player*

*Damian Kalita 58168 - Program, Klasa Exam, Klasa Biblioteka*

*Eliza Schwichtenberg 56858 – Program, Dokumentacja Techniczna, Klasa Work*

## Spis treści

|   |    |
|---|----|
| Spis treści.....                                      | 2  |
| 1. O co chodzi w grze? .....                          | 3  |
| 1.1 Elementy Gry .....                                | 3  |
| 1.2 Cel gry.....                                      | 3  |
| 1.3 Kiedy się wygrywa? Kiedy się przegrywa? .....     | 4  |
| 1.4 Co utrudnia grę? Co pomaga wygrać? .....          | 4  |
| 2. Architektura klas .....                            | 4  |
| 2.1 Opisz narzędzie, w którym wykonałeś schemat ..... | 4  |
| 2.2 Podstawowy schemat UML.....                       | 4  |
| 2.3 Relacje między obiektami.....                     | 4  |
| 2.4 Pełny schemat klas .....                          | 5  |
| 3. Kod programu.....                                  | 5  |
| 3.1 Narzędzie .....                                   | 5  |
| 3.2 Kod.....  | 6  |
| 3.2.1 Klasa Program.....                              | 6  |
| 3.2.2 Klasa Player .....                              | 12 |
| 3.2.3 Klasa Work.....                                 | 14 |
| 3.2.4 Klasa RandomEvent.....                          | 14 |
| 3.2.5 Klasa Exam .....                                | 21 |
| 3.2.6 Klasa Sklep .....                               | 34 |
| 4. Bibliografia .....                                 | 37 |

# 1. O co chodzi w grze?

## 1.1 Elementy Gry

Wcielamy się w studenta WSB. Zadaniem gracza jest przetrwanie 14 dni poprzedzających egzamin a następnie zdanie tego go. Gracz na start ma do dyspozycji następujące statystyki:

Morale (50)

Inteligencja (0)

Pieniądze (50)

Każdą ze statystyk można zwiększać kosztem innej. Na statystyki mają wpływ aktywności i zdarzenia losowe.

W skład aktywności wchodzi:

Biblioteka

Gracz może wybrać, ile godzin chce spędzić w bibliotece. Każda godzina skutkuje zwiększeniem inteligencji o 10 i zmniejszeniem morale o 10. Jeżeli poziom morali gracza spadnie poniżej 0, rozgrywka kończy się niepowodzeniem.

Praca

Gracz może wybrać, ile godzin pracuje. Każda godzina pracy skutkuje zwiększeniem ilości pieniędzy o 20zł i zmniejszeniem morali o 20. Jeżeli poziom morale gracza spadnie poniżej 0, rozgrywka kończy się niepowodzeniem.

Sklep

Wybierając sklep, gracz ma do wyboru 3 różne towary. Jednorazowo może wybrać jeden z nich. Każdy towar wpływa na statystyki w inny sposób. Do wyboru:

Zupka chińska (Morale +30, pieniądze -20)

Chleb (morale +60, pieniądze -35)

Piwo (morale +100, pieniądze -70)

Brak pieniędzy nie oznacza porażki w grze.

Po każdej aktywności mamy 33% szans na jedno z kilku przygotowanych zdarzeń losowych. Zdarzenia losowe mogą wpłynąć na statystyki pozytywnie lub negatywnie.

W grze został zaimplementowany system dobowy. W ciągu jednego dnia możemy wykonać jedną aktywność.

Gdy minie 14 dni gracz musi udać się na egzamin. Od tego jak trudne będą pytania zależy to, ile uzbierał inteligencji. Jeśli dobrze odpowiemy na przynajmniej 2 z 3 pytań wygrywamy, niższy wynik oznacza porażkę.

## 1.2 Cel gry

Celem gry jest przeżycie do momentu egzaminu i zdanie go.

### 1.3 Kiedy się wygrywa? Kiedy się przegrywa?

Wygrywa się, gdy minimum dwie odpowiedzi z trzech są poprawne. Przegrywa się, jeśli odpowie się na mniejszą ilość pytań lub nie dojdzie się do egzaminu.

### 1.4 Co utrudnia grę? Co pomaga wygrać?

Rozgrywkę mogą ułatwić bądź też utrudnić wydarzenia losowe. Czasami wynik zdarzenia losowego może być pozytywny lub negatywny, w zależności jaką decyzję podejmiemy.

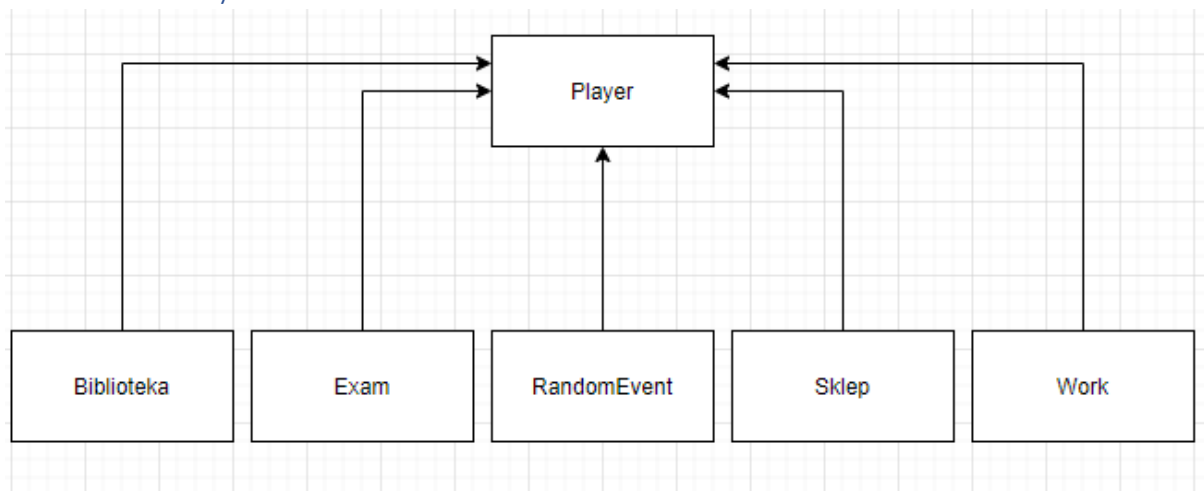
## 2. Architektura klas

### 2.1 Opisz narzędzie, w którym wykonałeś schemat

Schemat został wykonany z pomocą darmowego narzędzia online: <https://app.diagrams.net/>

Program jest intuicyjny oraz prosty w obsłudze, a dodatkowo nie wymaga uprzedniego zalogowania się na stronie internetowej.

### 2.2 Podstawowy schemat UML



### 2.3 Relacje między obiektami

**Biblioteka – Player :** klasa Biblioteka wykorzystuje pola 'intelligence' i 'morale' z klasy Player w celu zwiększenia oraz zmniejszenia ich.

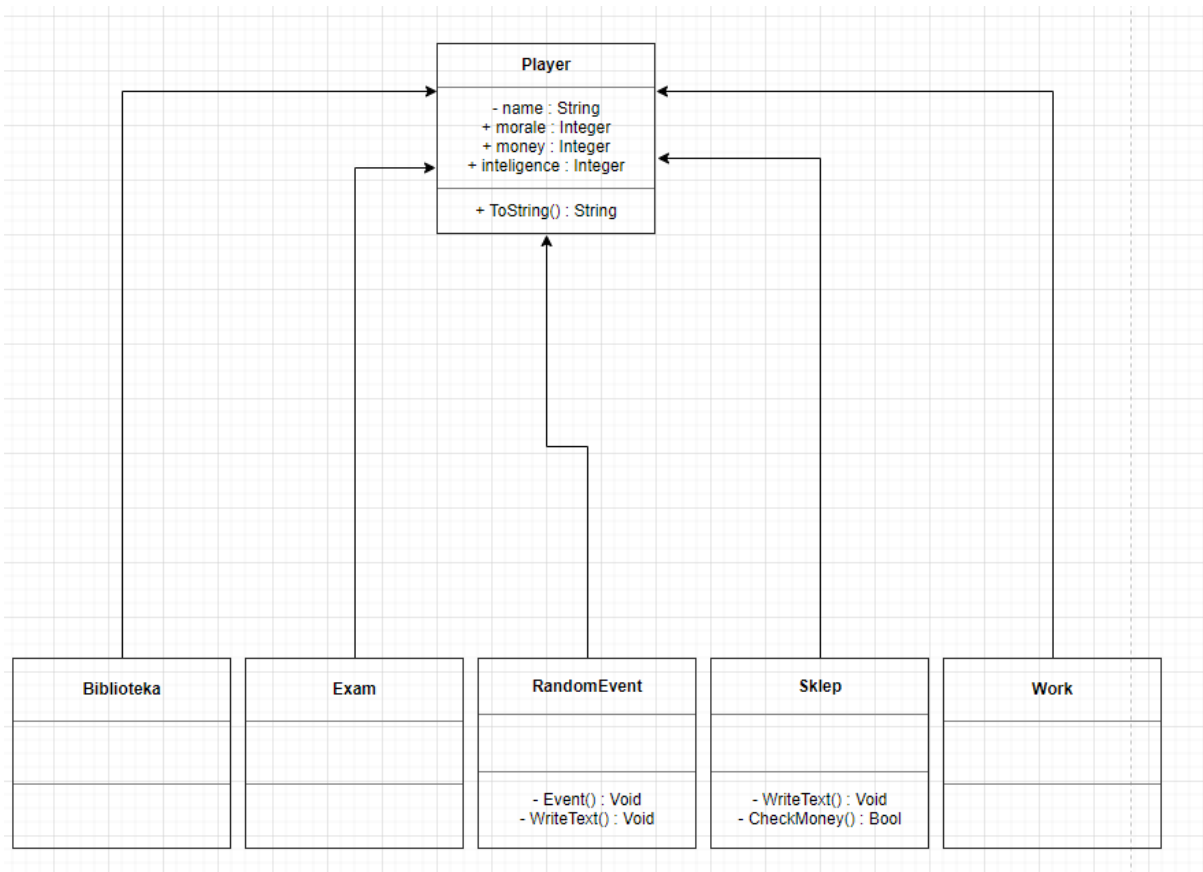
**Exam – Player :** klasa Exam wykorzystuje pole 'intelligence' z klasy Player w celu sprawdzenia jej wartości.

**RandomEvent – Player :** klasa RandomEvent wykorzystuje pola 'intelligence', 'morale' oraz 'money' z klasy Player w celu zwiększenia oraz zmniejszenia ich.

**Sklep – Player :** klasa Sklep wykorzystuje pola 'morale' oraz 'money' z klasy Player w celu zwiększenia oraz zmniejszenia ich.

**Work – Player :** klasa Work wykorzystuje pola 'money' i 'morale' z klasy Player w celu zwiększenia oraz zmniejszenia ich

## 2.4 Pełny schemat klas



## 3. Kod programu

### 3.1 Narzędzie

Gra została napisana w języku C# w programie Visual Studio 2019. Aby zagrać w grę należy pobrać pliki i uruchomić plik o nazwie "Projekt Zaliczeniowy Zarys" za pomocą Visual Studio 2019.

Projekt Zaliczeńowy Zarys Full\bin\Debug\netcoreapp3.1\ Projekt Zaliczeńowy Zarys

## 3.2 Kod

### 3.2.1 Klasa Program

```
using System;
using System.Threading;

namespace Projekt_Zaliczeńowy_Zarys
{

    class Program
    {

        static void RandomEventShow(int randomoption, int chances, Player
player)
        {
            //Ta funkcja odpowiada za to czy wydarzy się zdarzenie losowe
            if (randomoption <= chances)
            {
                RandomEvent randomEvent = new RandomEvent(player);
                Conf();
            }
        }

        static string Conf()
        {
            /*Ta funkcja wymusza na użytkownika wciśnięcie klawisza na
klawiaturze

            Jest ona potrzebna ponieważ po każdej akcji czyszczona jest
konsola a użytkownik musi zobaczyć wynik swojego działania*/
            Console.WriteLine("Wciśnij dowolny klawisz aby kontynuować");
            return Console.ReadLine();
        }
    }
}
```

```

static void WriteText(string text)
{
    //Ta funkcja odpowiada za wyświetlanie animowanego tekstu
    foreach(char c in text)
    {
        Console.Write(c);
        Thread.Sleep(10);
    }
}

static void Main(string[] args)
{
    string name;
    int option;
    int time;
    bool end = true;
    int day = 1;
    int examDay = 15;
    int daysToExam;
    Console.ForegroundColor = ConsoleColor.Cyan;

    WriteText("Podaj swoje imię\n");

    name = Convert.ToString(Console.ReadLine());

    Player player = new Player(name);

    WriteText("Do sesji pozostało tylko 14 dni, a ty w ogóle się
nie uczyłeś!\nJeżeli nie zdasz tego semestru rodzice wyrzucą cię z domu,
więc lepiej postaraj się przygotować!\n");
    Conf();
    while (end)
    {

```

Start:

```
        Console.ResetColor();
        Console.Clear();
        Console.ForegroundColor = ConsoleColor.Magenta;
        Console.WriteLine("|-----STUDENT
SIMULATOR-----|");
        Console.ResetColor();
        Console.ForegroundColor = ConsoleColor.Cyan;
        Console.WriteLine(player);
        Console.WriteLine("|-----
-----|");

        //To wbudowany zegar który odlicza dni do egzaminu
        daysToExam = examDay - day;

        if (daysToExam > 0) { Console.WriteLine("\nZostało " +
daysToExam + " dni do egzaminu!"); }
        else { Console.WriteLine("\nDzisiaj Egzamin!"); }

        //To menu podróży
        Console.WriteLine("\nGdzie chciałbyś się udać? \n 1:
Biblioteka \n 2: Praca \n 3: Sklep \n 4: Egzamin");
        Console.WriteLine("|-----
-----|");
        option = Convert.ToInt32(Console.ReadLine());

        //Ten fragment odpowiada za wylosowanie losowej liczby
potrzebnej do wydarzenia losowego
        Random random = new Random();
        int randomoption = random.Next(0, 101);

        switch (option)
        {

            //Case 1 odpowiada za opcje wybrania się do biblioteki
```



case 1:

```
if (daysToExam <= 0)
{
    WriteText("Musisz dzisiaj iść na egzamin!\n");
    Conf();
    goto Start;
}
```

```
WriteText("Jak długo zamierzasz się uczyć w
bibliotece? [1 godzina = -10 Morale +10 Inteligencja]\n");
```

```
time = Convert.ToInt32(Console.ReadLine());
```

```
Biblioteka biblioteka = new Biblioteka(time,
player);
```

```
RandomEventShow(randomoption, 30, player);
break;
```

case 2:

```
//Case 2 odpowiada za opcje wybrania się do do
pracy
```

```
if (daysToExam <= 0)
{
    WriteText("Musisz dzisiaj iść na egzamin!\n");
    Conf();
    goto Start;
}
```

```
WriteText("Jak długo zamierzasz pracować? [1
godzina = -15 Morale +19 Zł]\n");
```

```
time = Convert.ToInt32(Console.ReadLine());
```

```
Work work = new Work(time, player);
```

```

        RandomEventShow(randomoption, 30, player);
        break;
case 3:
    //Case 3 odpowiada za opcje wybrania się do sklepu
    if (daysToExam <= 0)
    {

        //Ten if uniemożliwi graczowi wybranie się w
        dane miejsce jeżeli musi udać się na egzamin
        WriteText("Musisz dzisiaj iść na egzamin!\n");
        Conf();

        goto Start;
    }

    Sklep sklep = new Sklep(player);

    RandomEventShow(randomoption, 30, player);
    break;
case 4:
    //Case 4 odpowiada za opcje wybrania się na egzamin
    if (daysToExam <= 0)
    {
        Exam exam = new Exam(player);
        end = false;
    }
    else
    {
        //ten if uniemożliwi udanie się egzamin jeżeli
        w dany dzień go nie ma
        WriteText("Dzisiaj nie ma egzaminu!\n");
        Conf();
    }

```

```

        goto Start;
    }
    break;
case 98:
    //Kod oszukujący pozwalający na zamknięcie gry
    end = false;
    break;
case 99:
    //Kod oszukujący pozwalający na wywołanie zdarzenia
losowego
    RandomEventShow(randomoption, 100, player);

    break;
default:
    //Ten fragment uniemożliwia wybranie innej opcji
niż te podane w menu i kody oszukujące
    WriteText("Wybierz poprawną opcje!\n");
    goto Start;
}

//W tym miejscu następuje postęp czasu
day++;

if (player.morale <= 0)
{
    //ekran końcowy w przypadku porażki wynikającej z zbyt
niskiego morale
    WriteText("Popadłeś w depresję, przegrywasz!\n");
    Conf();
    end = false;
}

```

```
//W tym miejscu normalizowane są wartości poszczególnych parametrów
klasy Player(Morale nie może przekroczyć 100, Pieniądze nie mogą spaść
poniżej 0)
```

```
        player.morale = Math.Min(player.morale, 100);
        player.money = Math.Max(player.money, 0);

    }
}
}
```

### 3.2.2 Klasa Player

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Projekt_Zaliczeńowy_Zarys
{
    class Player
    {
        private string name;
        public int morale;
        public int money;
        public int intelligence;
        //statystyki startowe
        public Player(string name)
        {
            this.name = name;
            money = 50;
            morale = 50;
        }
    }
}
```

```

        intelligence = 0;
    }

    //ekran eksponujący aktualne statystyki
    public override string ToString()
    {
        return $"Gracz:{name} \n Morale:{morale} \n
Inteligencja:{intelligence} \n Stan Konta:{money}";
    }
}

using System;
using System.Collections.Generic;
using System.Text;

namespace Projekt_Zaliczeńowy_Zarys
{
    class Player
    {
        private string name;
        public int morale;
        public int money;
        public int intelligence;
        //statystyki startowe
        public Player(string name)
        {
            this.name = name;
            money = 50;
            morale = 50;
            intelligence = 0;
        }
    }
}

```

```

        //ekran eksponujący aktualne statystyki
        public override string ToString()
        {
            return $"Gracz:{name} \n Morale:{morale} \n
Inteligencja:{intelligence} \n Stan Konta:{money}";
        }
    }
}

```

### 3.2.3 Klasa Work

```

using System;
using System.Collections.Generic;
using System.Text;

namespace Projekt_Zaliczeńowy_Zarys
{
    class Work
    {
        public Work(int time, Player player)
        {
            //pętla odpowiadająca za ilość godzin przebytych w pracy
            for (int i = 0; i < time; i++)
            {
                player.money += 20; //zwiększenie ilości pieniędzy
                player.morale -= 15; //zmniejszenie ilości morali
            }
        }
    }
}

```

### 3.2.4 Klasa RandomEvent

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Threading;

namespace Projekt_Zaliczeńowy_Zarys
{
    class RandomEvent
    {
        void Event(ref int value, int howMuch, bool plus)
        {
            //funkcja odpowiadająca za edycję poszczególnych parametrów
            klasy Player w zdarzeniu losowym
            if(plus) { value += howMuch; }
            else { value -= howMuch; }
        }
        //Ta funkcja odpowiada za wyświetlanie animowanego tekstu
        void WriteText(string text)
        {
            foreach (char c in text)
            {
                Console.Write(c);
                Thread.Sleep(10);
            }
        }

        public RandomEvent(Player player)
        {
            //Tu losowane jest jedno z 6 możliwych zdarzeń losowych
            Random random = new Random();
            int randomoption = random.Next(0, 6);
        }
    }
}

```

```

        switch (randomoption)
        {
            case 0:
                Console.Beep(659, 125); //komunikat dźwiękowy
                Console.Beep(659, 125);

                Console.WriteLine("|*****\n");

                WriteText("Napotkałeś wojownika ortalionu który zabrał
                ci portfel [-50zł]\n");

                Console.WriteLine("|*****\n");

                Event(ref player.money, 50, false);
                Event(ref player.morale, 20, false);
                break;

            case 1:
                Console.Beep(659, 125);
                Console.Beep(659, 125);

                Console.WriteLine("|*****\n");

                WriteText("Podejrzany typ w garniturze proponuje ci
                zainwestowanie twoich pieniędzy, zgadzasz się? [-50zł]\n");

                WriteText("1. Tak \n2. Nie\n");

                Console.WriteLine("|*****\n");

                int option = Convert.ToInt32(Console.ReadLine());
                switch (option)
                {
                    case 1:
                        Event(ref player.money, 50, false);
                        int invest = random.Next(2);
                        if (invest == 0) { Event(ref player.money, 100,
true);

```



```
Console.WriteLine("|*****\n");
```

```
WriteText("Inwestycja się udała, twój  
kapitał został pomnożony o 100%\n");
```

```
Console.WriteLine("|*****\n"); }
```

```
else if (invest != 0) {
```

```
Console.WriteLine("|*****\n");
```

```
WriteText("Inwestycja się nie powiodła,  
straciłeś swój kapitał\n");
```

```
Console.WriteLine("|*****\n"); }
```

```
break;
```

```
case 2:
```

```
break;
```

```
default:
```

```
break;
```

```
}
```

```
break;
```

```
case 2:
```

```
Console.Beep(659, 125);
```

```
Console.Beep(659, 125);
```

```
Console.WriteLine("|*****\n");
```

```
WriteText("Kiedy wracałeś do domu, twój autobus  
odjechał ci przed nosem, a następny dopiero za godzinę [-20 Morale] \n");
```

```
Console.WriteLine("|*****\n");
```

```

        Event(ref player.morale, 20, false);
        break;
    case 3:
        Console.Beep(659, 125);
        Console.Beep(659, 125);

Console.WriteLine("|*****\n");

        WriteText("Napotykasz pod sklepem starego mędrca, który
za jedynie dwa złote oferuje ci podzielenie się swoją wiedzą, akceptujesz?
[+10 Inteligencji -2 zł]\n");

        WriteText("1. Tak \n2. Nie\n");

Console.WriteLine("|*****\n");

        int option1 = Convert.ToInt32(Console.ReadLine());
        switch (option1)
        {
            case 1:
                Event(ref player.intelligence, 10, true);
                Event(ref player.money, 2, false);
                break;
            case 2:
                break;
            default:
                break;
        }
        break;
    case 4:
        Console.Beep(659, 125);
        Console.Beep(659, 125);

Console.WriteLine("|*****\n");

```

```
        WriteText("Na swojej drodze napotykasz youtubera  
nagrywającego pranka. Po tym co zobaczyłeś od razu czujesz się głupszy [-20  
Inteligencji]\n");
```

```
Console.WriteLine("|*****  
*****|");
```

```
        Event(ref player.intelligence, 20, false);
```

```
        break;
```

```
    case 5:
```

```
        Console.Beep(659, 125);
```

```
        Console.Beep(659, 125);
```

```
Console.WriteLine("|*****  
*****|\n");
```

```
        WriteText("Napotykasz dżentelmena, który proponuje ci  
napad na hurtownię tanich wyrobów alkoholowych, zgadasz się? [???]\n");
```

```
        WriteText("1. Tak \n2. Nie\n");
```

```
Console.WriteLine("|*****  
*****|");
```

```
        int option2 = Convert.ToInt32(Console.ReadLine());
```

```
        switch (option2)
```

```
        {
```

```
            case 1:
```

```
                int heist = random.Next(1);
```

```
                if (heist == 0)
```

```
                {
```

```
Console.WriteLine("|*****  
*****|\n");
```

```
        WriteText("Skok zakończył się pomyślnie,  
otrzymujesz swoją część z zarobku [+100 zł]\n");
```

```
Console.WriteLine("|*****  
*****|");
```

```
        Event(ref player.money, 100, true);
```

```
    }
```

```

        else if (heist == 1)
        {

Console.WriteLine("|*****\n");

                WriteText("Napad zakończył się
niepowodzeniem, ponieważ Polonez którym uciekaliście z hurtowni uległ
awarii" +

                        "\n Na szczęście wartość skradzionych
dóbr była na tak niska, że skończyło się tylko na grzywnie [-20 Morale, -
100 zł]\n");

Console.WriteLine("|*****\n");

                Event(ref player.money, 100, false);
                Event(ref player.morale, 20, false);
        }
        break;
    case 2:
        break;
    default:
        break;
    }
    break;
case 6:
    {
        Console.Beep(659, 125);
        Console.Beep(659, 125);

Console.WriteLine("|*****\n");

                WriteText("Idąc chodnikiem, zauważasz na nim coś
świeącego. Okazuje się że to 5 złotych [+20 Morale, +5zł]\n");

Console.WriteLine("|*****\n");

                Event(ref player.money, 5, true);

```

```

        Event(ref player.morale, 20, true);
        break;
    }

    }

}

}
}
}

```

### 3.2.5 Klasa Exam

```

using System;
using System.Collections.Generic;
using System.Text;

using System.Threading;

namespace Projekt_Zaliczeniowy_Zarys
{
    class Exam
        //egzamin skonstutowany jest tak aby pytania były zależne od
        //inteligencji. Im mniejsza inteligencja tym trudniejsze pytania
    {

        public Exam(Player player)
        {
            int option = 0; //wersja testu
            int answer; //wybrana odpowiedź na pytanie
            int correctanswers = 0; //licznik poprawnych odpowiedzi
            //warunek który ustala wersję egzaminu

```

```

    if (player.intelligence <= 10)
    {
        option = 0;
    }
    else if (player.intelligence <= 50)
    {
        option = 1;
    }
    else if (player.intelligence <= 80)
    {
        option = 2;
    }
    else if (player.intelligence >= 100)
    {
        option = 3;
    }
    switch (option)
    {
        //najtrudniejszy zestaw pytań
        case 0:
            Console.WriteLine("|-----
            -----|\n");

            Console.WriteLine("Kto wprowadził znak równości (=)
do zapisu matematycznego?\n");

            Console.WriteLine("1. Euklides z Aleksandrii");
            Console.WriteLine("2. Simon Stevin");
            Console.WriteLine("3. Robert Rercorde\n");
            Console.WriteLine("Wybierz poprawną odpowiedź (1, 2 lub
3)");

            Console.WriteLine("|-----
            -----|");

            answer = Convert.ToInt32(Console.ReadLine());
//odebranie odpowiedzi od gracza

            if (answer == 3)

```

```

        {
            correctanswers++; //zwiększenie licznika poprawnych
odpowiedzi
        }
        Console.WriteLine("|-----|
        -----|\n");
        Console.WriteLine("Jeden centylion - co to za
liczba?\n");

        Console.WriteLine("1. 10 do potęgi 42 ");
        Console.WriteLine("2. 10 do potęgi 600");
        Console.WriteLine("3. 10 do potęgi 1024\n");
        Console.WriteLine("Wybierz poprawną odpowiedź (1, 2 lub
3)");

        Console.WriteLine("|-----|
        -----|");
        answer = Convert.ToInt32(Console.ReadLine());
        if (answer == 2)
        {
            correctanswers++;
        }
        Console.WriteLine("|-----|
        -----|\n");
        Console.WriteLine("Za pomocą zasady Cavalieriego
możemy:\n");

        Console.WriteLine("1. Porównywać objętość brył");
        Console.WriteLine("2. Znaleźć wszystkie liczby pierwsze
w zadanym przedziale liczbowym");
        Console.WriteLine("3. Szukać liczb pierwszych
mniejszych od 2971\n");
        Console.WriteLine("Wybierz poprawną odpowiedź (1, 2 lub
3)");

        Console.WriteLine("|-----|
        -----|");
        answer = Convert.ToInt32(Console.ReadLine());
        if (answer == 1)
        {

```

```

        correctanswers++;
    }
    break;
case 1:
    Console.WriteLine("|-----
-----|\n");
    Console.WriteLine("Gdzie urodził się Fryderyk
Chopin?\n");
    Console.WriteLine("1. W Żelazowej woli");
    Console.WriteLine("2. W Paryżu");
    Console.WriteLine("3. Na Honolulu\n");
    Console.WriteLine("Wybierz poprawną odpowiedź (1, 2 lub
3)");
    Console.WriteLine("|-----
-----|");
    answer = Convert.ToInt32(Console.ReadLine());
    if (answer == 1)
    {
        correctanswers++;
    }
    Console.WriteLine("|-----
-----|\n");
    Console.WriteLine("Kto był autorem Iliady i
Odysei?\n");
    Console.WriteLine("1. Wisława Szymborska");
    Console.WriteLine("2. Homer");
    Console.WriteLine("3. J.K. Rowling\n");
    Console.WriteLine("Wybierz poprawną odpowiedź (1, 2 lub
3)");
    Console.WriteLine("|-----
-----|");
    answer = Convert.ToInt32(Console.ReadLine());
    if (answer == 2)
    {
        correctanswers++;
    }

```



```

    }
    Console.WriteLine("|-----
-----|\n");
    Console.WriteLine("Kto rządzi Polską?\n");
    Console.WriteLine("1. Andrzej Duda");
    Console.WriteLine("2. Jarosław Kaczyński");
    Console.WriteLine("3. Mateusz Morawiecki\n");
    Console.WriteLine("Wybierz poprawną odpowiedź (1, 2 lub
3)");

    Console.WriteLine("|-----
-----|");

    answer = Convert.ToInt32(Console.ReadLine());
    if (answer == 2)
    {
        correctanswers++;
    }
    break;
    //test o średniej trudności
case 2:
    Console.WriteLine("|-----
-----|\n");
    Console.WriteLine("Mikołaj Kopernik:\n");
    Console.WriteLine("1. Wyszedł po mleko i nie wrócił");
    Console.WriteLine("2. Był pierwszym człowiekiem w
kosmosie");
    Console.WriteLine("3. Wstrzymał Słońce, ruszył
Ziemię\n");
    Console.WriteLine("Wybierz poprawną odpowiedź (1, 2 lub
3)");

    Console.WriteLine("|-----
-----|");

    answer = Convert.ToInt32(Console.ReadLine());
    if (answer == 3)
    {

```

```

        correctanswers++;
    }
    Console.WriteLine("|-----
-----|\n");
    Console.WriteLine("Ile było rozbiorów Polski?\n");
    Console.WriteLine("1. Jeden");
    Console.WriteLine("2. Trzy");
    Console.WriteLine("3. Osiem\n");
    Console.WriteLine("Wybierz poprawną odpowiedź (1, 2 lub
3)");
    Console.WriteLine("|-----
-----|");
    answer = Convert.ToInt32(Console.ReadLine());
    if (answer == 2)
    {
        correctanswers++;
    }
    Console.WriteLine("|-----
-----|\n");
    Console.WriteLine("Jak nazywano królów starożytnych
egipcjan?\n");
    Console.WriteLine("1. Faraon");
    Console.WriteLine("2. Prezydent");
    Console.WriteLine("3. Dyktator\n");
    Console.WriteLine("Wybierz poprawną odpowiedź (1, 2 lub
3)");
    Console.WriteLine("|-----
-----|");
    answer = Convert.ToInt32(Console.ReadLine());
    if (answer == 1)
    {
        correctanswers++;
    }
    break;

```

```

//najłatwiejszy test
case 3:
    Console.WriteLine("|-----
-----|\n");
    Console.WriteLine("Ile jest 2+2?\n");
    Console.WriteLine("1. Osiem");
    Console.WriteLine("2. Czternaście");
    Console.WriteLine("3. Cztery\n");
    Console.WriteLine("Wybierz poprawną odpowiedź (1, 2 lub
3)");
    Console.WriteLine("|-----
-----|");
    answer = Convert.ToInt32(Console.ReadLine());
    if (answer == 3)
    {
        correctanswers++;
    }
    Console.WriteLine("|-----
-----|\n");
    Console.WriteLine("Kto skonstruował pierwszy
telefon?\n");
    Console.WriteLine("1. Aleksander Bell");
    Console.WriteLine("2. Nicola Tesla ");
    Console.WriteLine("3. Albert Einstein\n");
    Console.WriteLine("Wybierz poprawną odpowiedź (1, 2 lub
3)");
    Console.WriteLine("|-----
-----|");

    answer = Convert.ToInt32(Console.ReadLine());
    if (answer == 1)
    {
        correctanswers++;

```

```

    }
    Console.WriteLine("|-----
-----|\n");

    Console.WriteLine("Jedno piwo to wstęp do trzech, a
trzy to wstęp do: ?\n");

    Console.WriteLine("1. Dziewięciu");
    Console.WriteLine("2. Dwudziestu");
    Console.WriteLine("3. Pięciu\n");
    Console.WriteLine("Wybierz poprawną odpowiedź (1, 2 lub
3)");

    Console.WriteLine("|-----
-----|");

    answer = Convert.ToInt32(Console.ReadLine());
    if (answer == 1)
    {
        correctanswers++;
    }

    break;
}
//podsumowanie egzaminu
Console.WriteLine("Podsumowanie:");
if (correctanswers >= 2)
{
    //komunikat w przypadku zdania egzaminu
    Console.WriteLine("Zdobyte punkty: " + correctanswers);
    Console.WriteLine("UDAŁO SIĘ! ZDAŁEŚ EGZAMIN!");
}
else
{
    //komunikat w przypadku niezdania egzaminu
    Console.WriteLine("Zdobyte punkty: " + correctanswers);
    Console.WriteLine("NIE ZDAŁEŚ EGZAMINU! KONIEC GRY!");
}

```

```
    }  
    //muzyka na koniec gry (motyw Mario)  
    Console.Beep(659, 125);  
    Console.Beep(659, 125);  
    Thread.Sleep(125);  
    Console.Beep(659, 125);  
    Thread.Sleep(167);  
    Console.Beep(523, 125);  
    Console.Beep(659, 125);  
    Thread.Sleep(125);  
    Console.Beep(784, 125);  
    Thread.Sleep(375);  
    Console.Beep(392, 125);  
    Thread.Sleep(375);  
    Console.Beep(523, 125);  
    Thread.Sleep(250);  
    Console.Beep(392, 125);  
    Thread.Sleep(250);  
    Console.Beep(330, 125);  
    Thread.Sleep(250);  
    Console.Beep(440, 125);  
    Thread.Sleep(125);  
    Console.Beep(494, 125);  
    Thread.Sleep(125);  
    Console.Beep(466, 125);  
    Thread.Sleep(42);  
    Console.Beep(440, 125);  
    Thread.Sleep(125);  
    Console.Beep(392, 125);  
    Thread.Sleep(125);  
    Console.Beep(659, 125);
```

```
Thread.Sleep(125);
Console.Beep(784, 125);
Thread.Sleep(125);
Console.Beep(880, 125);
Thread.Sleep(125);
Console.Beep(698, 125);
Console.Beep(784, 125);
Thread.Sleep(125);
Console.Beep(659, 125);
Thread.Sleep(125);
Console.Beep(523, 125);
Thread.Sleep(125);
Console.Beep(587, 125);
Console.Beep(494, 125);
Thread.Sleep(125);
Console.Beep(523, 125);
Thread.Sleep(250);
Console.Beep(392, 125);
Thread.Sleep(250);
Console.Beep(330, 125);
Thread.Sleep(250);
Console.Beep(440, 125);
Thread.Sleep(125);
Console.Beep(494, 125);
Thread.Sleep(125);
Console.Beep(466, 125);
Thread.Sleep(42);
Console.Beep(440, 125);
Thread.Sleep(125);
Console.Beep(392, 125);
Thread.Sleep(125);
Console.Beep(659, 125);
```

```
Thread.Sleep(125);
Console.Beep(784, 125);
Thread.Sleep(125);
Console.Beep(880, 125);
Thread.Sleep(125);
Console.Beep(698, 125);
Console.Beep(784, 125);
Thread.Sleep(125);
Console.Beep(659, 125);
Thread.Sleep(125);
Console.Beep(523, 125);
Thread.Sleep(125);
Console.Beep(587, 125);
Console.Beep(494, 125);
Thread.Sleep(375);
Console.Beep(784, 125);
Console.Beep(740, 125);
Console.Beep(698, 125);
Thread.Sleep(42);
Console.Beep(622, 125);
Thread.Sleep(125);
Console.Beep(659, 125);
Thread.Sleep(167);
Console.Beep(415, 125);
Console.Beep(440, 125);
Console.Beep(523, 125);
Thread.Sleep(125);
Console.Beep(440, 125);
Console.Beep(523, 125);
Console.Beep(587, 125);
Thread.Sleep(250);
Console.Beep(784, 125);
```

```
Console.Beep(740, 125);
Console.Beep(698, 125);
Thread.Sleep(42);
Console.Beep(622, 125);
Thread.Sleep(125);
Console.Beep(659, 125);
Thread.Sleep(167);
Console.Beep(698, 125);
Thread.Sleep(125);
Console.Beep(698, 125);
Console.Beep(698, 125);
Thread.Sleep(625);
Console.Beep(784, 125);
Console.Beep(740, 125);
Console.Beep(698, 125);
Thread.Sleep(42);
Console.Beep(622, 125);
Thread.Sleep(125);
Console.Beep(659, 125);
Thread.Sleep(167);
Console.Beep(415, 125);
Console.Beep(440, 125);
Console.Beep(523, 125);
Thread.Sleep(125);
Console.Beep(440, 125);
Console.Beep(523, 125);
Console.Beep(587, 125);
Thread.Sleep(250);
Console.Beep(622, 125);
Thread.Sleep(250);
Console.Beep(587, 125);
Thread.Sleep(250);
```



```
Console.Beep(523, 125);
Thread.Sleep(1125);
Console.Beep(784, 125);
Console.Beep(740, 125);
Console.Beep(698, 125);
Thread.Sleep(42);
Console.Beep(622, 125);
Thread.Sleep(125);
Console.Beep(659, 125);
Thread.Sleep(167);
Console.Beep(415, 125);
Console.Beep(440, 125);
Console.Beep(523, 125);
Thread.Sleep(125);
Console.Beep(440, 125);
Console.Beep(523, 125);
Console.Beep(587, 125);
Thread.Sleep(250);
Console.Beep(784, 125);
Console.Beep(740, 125);
Console.Beep(698, 125);
Thread.Sleep(42);
Console.Beep(622, 125);
Thread.Sleep(125);
Console.Beep(659, 125);
Thread.Sleep(167);
Console.Beep(698, 125);
Thread.Sleep(125);
Console.Beep(698, 125);
Console.Beep(698, 125);
Thread.Sleep(625);
Console.Beep(784, 125);
```

```

Console.Beep(740, 125);
Console.Beep(698, 125);
Thread.Sleep(42);
Console.Beep(622, 125);
Thread.Sleep(125);
Console.Beep(659, 125);
Thread.Sleep(167);
Console.Beep(415, 125);
Console.Beep(440, 125);
Console.Beep(523, 125);
Thread.Sleep(125);
Console.Beep(440, 125);
Console.Beep(523, 125);
Console.Beep(587, 125);
Thread.Sleep(250);
Console.Beep(622, 125);
Thread.Sleep(250);
Console.Beep(587, 125);
Thread.Sleep(250);
Console.Beep(523, 125);

        Console.WriteLine("Made by:\n Eliza Schwichtenberg \n Szymon
Paluszewski \n Bartosz Suska \n Damian Kalita");
    }
}
}

```

### 3.2.6 Klasa Sklep

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Threading;

```

```
namespace Projekt_Zaliczeńowy_Zarys
{
    class Sklep
    {
        //animacja tekstu
        void WriteText(string text)
        {
            foreach (char c in text)
            {
                Console.Write(c);
                Thread.Sleep(10);
            }
        }
        //sprawdzenie wymaganej ilości pieniędzy
        bool CheckMoney(int money, int cost)
        {
            if(money < cost)
            {
                WriteText("Nie masz tyle pieniędzy\n");
                return false;
            }
            return true;
        }
        //menu sklepu
        public Sklep(Player player)
        {
            Start:
            WriteText("Co chcesz kupić?\n");
            WriteText("1: Zupka Chińska[Morale +30 Cena 20zł]\n");
            WriteText("2: Chleb[Morale +60 Cena 35zł]\n");
            WriteText("3: Piwo[Morale +100 Cena 70zł]\n");
            WriteText("4: Wyjście ze sklepu\n");
        }
    }
}
```

```

        int userInput = Convert.ToInt32(Console.ReadLine());
//zmienna przechowująca wybraną opcję w sklepie

        switch (userInput)
        {
            case 1:
                if(CheckMoney(player.money, 20)) //sprawdzenie
wymaganej ilości pieniędzy
                {
                    //zamiana statytek w zależności od opcji
                    player.morale += 30;
                    player.money -= 20;
                }
                else
                {
                    WriteText("Wybierz coś innego\n"); //komunikat w
przypadku braku wymaganej kwoty
                }

                break;
            case 2:
                if (CheckMoney(player.money, 35))
                {
                    player.morale += 60;
                    player.money -= 35;
                }
                else
                {
                    WriteText("Wybierz coś innego\n");
                }

                break;
            case 3:

```

```

        if (CheckMoney(player.money, 70))
        {
            player.morale += 100;
            player.money -= 70;
        }
        else
        {
            WriteText("Wybierz coś innego\n");
        }

        break;
    case 4:
        WriteText("Wychodzisz ze sklepu\n"); //komunikat przy
        wybraniu opcji wyjścia
        break;
    default:
        WriteText("Wybierz poprawną opcje!\n"); //komunikat w
        przypadku wybrania nieistniejącej opcji
        goto Start;

    }
}
}
}
}

```

## 4. Bibliografia

<https://docs.microsoft.com/pl-pl/dotnet/csharp/>

<https://visualstudio.microsoft.com/pl/vs/>

<https://hashtagakash.wordpress.com/2014/01/22/182/>

VS2019

<https://app.diagrams.net/> (draw.io)