

dokumentacjaDoxy

Generated by Doxygen 1.9.6



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Class Documentation</b>	<b>7</b>
4.1 Bilet Class Reference	7
4.1.1 Detailed Description	7
4.1.2 Constructor & Destructor Documentation	8
4.1.2.1 Bilet()	8
4.1.3 Member Function Documentation	8
4.1.3.1 getCena()	8
4.1.3.2 getTyp()	8
4.1.3.3 odczytajWszystkieZPliku()	8
4.2 Event Class Reference	9
4.2.1 Detailed Description	10
4.2.2 Constructor & Destructor Documentation	10
4.2.2.1 Event()	10
4.2.3 Member Function Documentation	10
4.2.3.1 getCzas()	10
4.2.3.2 getData()	10
4.2.3.3 getNazwa()	11
4.2.3.4 odczytajWszystkieZPliku()	11
4.2.4 Friends And Related Function Documentation	11
4.2.4.1 operator<<	11
4.3 Gad Class Reference	12
4.3.1 Detailed Description	13
4.3.2 Constructor & Destructor Documentation	13
4.3.2.1 Gad()	14
4.3.3 Member Function Documentation	14
4.3.3.1 getDlCiala()	14
4.3.3.2 getJadowite()	14
4.3.3.3 getSposobRuchu()	15
4.3.3.4 pokazInfo()	15
4.4 Interfejs Class Reference	15
4.4.1 Detailed Description	16
4.4.2 Member Function Documentation	16
4.4.2.1 menu()	16
4.4.2.2 menuPersonel()	17

4.4.2.3 menuUzytkownik()	17
4.4.2.4 usunBilet()	17
4.4.2.5 usunEvent()	17
4.4.2.6 usunZwierze()	18
4.4.2.7 zapiszBilet()	18
4.5 Personel Class Reference	18
4.5.1 Detailed Description	19
4.5.2 Constructor & Destructor Documentation	19
4.5.2.1 Personel()	19
4.6 Plaz Class Reference	19
4.6.1 Detailed Description	20
4.6.2 Constructor & Destructor Documentation	21
4.6.2.1 Plaz()	21
4.6.3 Member Function Documentation	21
4.6.3.1 getKolorSkory()	21
4.6.3.2 getSpRozmnazania()	22
4.6.3.3 getWodne()	22
4.6.3.4 pokazInfo()	22
4.7 Ptak Class Reference	22
4.7.1 Detailed Description	24
4.7.2 Constructor & Destructor Documentation	24
4.7.2.1 Ptak()	24
4.7.3 Member Function Documentation	24
4.7.3.1 getLatajacy()	24
4.7.3.2 getMigracje()	25
4.7.3.3 getRodzajGniazda()	25
4.7.3.4 getRozpietoscSkrzydel()	25
4.7.3.5 pokazInfo()	25
4.8 Ryba Class Reference	26
4.8.1 Detailed Description	27
4.8.2 Constructor & Destructor Documentation	27
4.8.2.1 Ryba()	27
4.8.3 Member Function Documentation	28
4.8.3.1 getDlugosc()	28
4.8.3.2 getJajorodne()	28
4.8.3.3 getRodzajSkrzeli()	28
4.8.3.4 pokazInfo()	29
4.8.3.5 setDlugosc()	29
4.8.3.6 setJajorodne()	29
4.8.3.7 setRodzajSkrzeli()	29
4.9 Ssak Class Reference	30
4.9.1 Detailed Description	31

4.9.2 Constructor & Destructor Documentation	31
4.9.2.1 Ssak()	31
4.9.3 Member Function Documentation	32
4.9.3.1 getLiczbaPazurow()	32
4.9.3.2 getTrybZycia()	32
4.9.3.3 getZagrozony()	32
4.9.3.4 pokazInfo()	32
4.10 Zoo Class Reference	33
4.11 Zwiedzajacy Class Reference	33
4.11.1 Detailed Description	33
4.11.2 Constructor & Destructor Documentation	33
4.11.2.1 Zwiedzajacy()	33
4.11.3 Member Function Documentation	34
4.11.3.1 getImie()	34
4.11.3.2 getNazwisko()	34
4.11.3.3 getNrTel()	34
4.12 Zwierze Class Reference	35
4.12.1 Detailed Description	35
4.12.2 Constructor & Destructor Documentation	36
4.12.2.1 Zwierze()	36
4.12.3 Member Function Documentation	36
4.12.3.1 getDieta()	36
4.12.3.2 getDlZycia()	36
4.12.3.3 getGatunek()	37
4.12.3.4 getImie()	37
4.12.3.5 getPochodzenie()	37
4.12.3.6 getSezon()	37
4.12.3.7 pokazInfo()	37
<b>5 File Documentation</b>	<b>39</b>
5.1 Bilet.h	39
5.2 Event.h	39
5.3 Gad.h	40
5.4 Interfejs.h	40
5.5 Personel.h	41
5.6 Plaz.h	41
5.7 Ptak.h	42
5.8 Ryba.h	42
5.9 Ssak.h	42
5.10 Zoo.h	43
5.11 Zwiedzajacy.h	44
5.12 Zwierze.h	44



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Bilet . . . . .	7
Event . . . . .	9
Interfejs . . . . .	15
Personel . . . . .	18
Zoo . . . . .	33
Zwiedzajacy . . . . .	33
Zwierze . . . . .	35
Gad . . . . .	12
Plaz . . . . .	19
Ptak . . . . .	22
Ryba . . . . .	26
Ssak . . . . .	30





## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Bilet</a>	Reprezentuje bilet . . . . .	<a href="#">7</a>
<a href="#">Event</a>	Reprezentuje wydarzenie . . . . .	<a href="#">9</a>
<a href="#">Gad</a>	Reprezentuje gadzi zwierzę . . . . .	<a href="#">12</a>
<a href="#">Interfejs</a>	Klasa reprezentująca interfejs aplikacji . . . . .	<a href="#">15</a>
<a href="#">Personel</a>	Klasa reprezentująca personel obsługujący aplikację . . . . .	<a href="#">18</a>
<a href="#">Plaz</a>	Klasa reprezentująca płazy . . . . .	<a href="#">19</a>
<a href="#">Ptak</a>	Klasa reprezentująca ptaki . . . . .	<a href="#">22</a>
<a href="#">Ryba</a>	Klasa reprezentująca ryby . . . . .	<a href="#">26</a>
<a href="#">Ssak</a>	Klasa reprezentująca ssaki . . . . .	<a href="#">30</a>
<a href="#">Zoo</a>	. . . . .	<a href="#">33</a>
<a href="#">Zwiedzający</a>	Klasa reprezentująca zwiedzającego . . . . .	<a href="#">33</a>
<a href="#">Zwierze</a>	Klasa przechowująca informacje o zwierzętach . . . . .	<a href="#">35</a>



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">Bilet.h</a>	??
<a href="#">Event.h</a>	??
<a href="#">Gad.h</a>	??
<a href="#">Interfejs.h</a>	??
<a href="#">Personel.h</a>	??
<a href="#">Plaz.h</a>	??
<a href="#">Ptak.h</a>	??
<a href="#">Ryba.h</a>	??
<a href="#">Ssak.h</a>	??
<a href="#">Zoo.h</a>	??
<a href="#">Zwiedzajacy.h</a>	??
<a href="#">Zwierze.h</a>	??



## Chapter 4

# Class Documentation

### 4.1 Bilet Class Reference

Reprezentuje bilet.

```
#include <Bilet.h>
```

#### Public Member Functions

- **Bilet** ()  
*Konstruktor domyślny.*
- **Bilet** (const std::string &typ, const double &cena)  
*Konstruktor parametryczny.*
- **~Bilet** ()  
*Destruktor.*
- std::string **getTyp** () const  
*Pobiera typ biletu.*
- double **getCena** () const  
*Pobiera cenę biletu.*
- virtual void **pokazInfo** () const  
*Wyświetla informacje o bilecie.*

#### Static Public Member Functions

- static std::vector< **Bilet** > **odczytajWszystkieZPliku** (const std::string &nazwapliku)  
*Odczytuje wszystkie bilety z pliku.*

#### 4.1.1 Detailed Description

Reprezentuje bilet.

## 4.1.2 Constructor & Destructor Documentation

### 4.1.2.1 Bilet()

```
Bilet::Bilet (
    const std::string & typ,
    const double & cena )
```

Konstruktor parametryczny.

#### Parameters

<i>typ</i>	Typ biletu.
<i>cena</i>	Cena biletu.

## 4.1.3 Member Function Documentation

### 4.1.3.1 getCena()

```
double Bilet::getCena ( ) const
```

Pobiera cenę biletu.

#### Returns

Cena biletu.

### 4.1.3.2 getTyp()

```
std::string Bilet::getTyp ( ) const
```

Pobiera typ biletu.

#### Returns

Typ biletu.

### 4.1.3.3 odczytajWszystkieZPliku()

```
std::vector< Bilet > Bilet::odczytajWszystkieZPliku (
    const std::string & nazwapliku ) [static]
```

Odczytuje wszystkie bilety z pliku.

## Parameters

<i>nazwapliku</i>	Nazwa pliku.
-------------------	--------------

## Returns

Wektor zawierający wszystkie bilety odczytane z pliku.

The documentation for this class was generated from the following files:

- Bilet.h
- Bilet.cpp

## 4.2 Event Class Reference

Reprezentuje wydarzenie.

```
#include <Event.h>
```

### Public Member Functions

- **Event** ()  
*Konstruktor domyślny.*
- **Event** (const std::string &nazwa, const std::string &czas, const std::string &data)  
*Konstruktor parametryczny.*
- **~Event** ()  
*Destruktor.*
- std::string **getNazwa** () const  
*Pobiera nazwę wydarzenia.*
- std::string **getCzas** () const  
*Pobiera czas wydarzenia.*
- std::string **getData** () const  
*Pobiera datę wydarzenia.*
- void **godzinyOtwarcia** ()  
*Wyświetla godziny otwarcia dla danego wydarzenia.*

### Static Public Member Functions

- static std::vector< **Event** > **odczytajWszystkieZPliku** (const std::string &nazwapliku)  
*Odczytuje wszystkie wydarzenia z pliku.*

### Friends

- std::ostream & **operator<<** (std::ostream &os, const **Event** &event)  
*Przeciążenie operatora strumieniowego dla wyjścia.*

### 4.2.1 Detailed Description

Reprezentuje wydarzenie.

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 Event()

```
Event::Event (
    const std::string & nazwa,
    const std::string & czas,
    const std::string & data )
```

Konstruktor parametryczny.

##### Parameters

<i>nazwa</i>	Nazwa wydarzenia.
<i>czas</i>	Czas wydarzenia.
<i>data</i>	Data wydarzenia.

### 4.2.3 Member Function Documentation

#### 4.2.3.1 getCzas()

```
std::string Event::getCzas ( ) const
```

Pobiera czas wydarzenia.

##### Returns

Czas wydarzenia.

#### 4.2.3.2 getData()

```
std::string Event::getData ( ) const
```

Pobiera datę wydarzenia.

##### Returns

Data wydarzenia.



#### 4.2.3.3 getNazwa()

```
std::string Event::getNazwa ( ) const
```

Pobiera nazwę wydarzenia.

##### Returns

Nazwa wydarzenia.

#### 4.2.3.4 odczytajWszystkieZPliku()

```
std::vector< Event > Event::odczytajWszystkieZPliku (
    const std::string & nazwapliku ) [static]
```

Odczytuje wszystkie wydarzenia z pliku.

##### Parameters

<i>nazwapliku</i>	Nazwa pliku.
-------------------	--------------

##### Returns

Wektor zawierający wszystkie wydarzenia odczytane z pliku.

### 4.2.4 Friends And Related Function Documentation

#### 4.2.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const Event & event ) [friend]
```

Przeciążenie operatora strumieniowego dla wyjścia.

Przeciążony operator << umożliwia wygodne wyświetlanie obiektów klasy [Event](#) na strumieniu wyjściowym, takim jak `std::cout` czy strumień plikowy.

##### Parameters

<i>os</i>	Referencja do strumienia wyjściowego.
<i>event</i>	Referencja do obiektu klasy <a href="#">Event</a> , który ma być wyświetlony.

## Returns

Referencja do strumienia wyjściowego po operacji wyświetlenia.

The documentation for this class was generated from the following files:

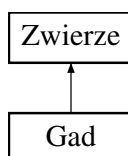
- Event.h
- Event.cpp

## 4.3 Gad Class Reference

Reprezentuje gady zwierzę.

```
#include <Gad.h>
```

Inheritance diagram for Gad:



## Public Member Functions

- **Gad ()**  
*Konstruktor domyślny.*
- **Gad** (const std::string &, const std::string &, const std::string &, const std::string &, const int &, const std::string &, const bool &, const std::string &, const int &)  
*Konstruktor inicjalizujący właściwości gada.*
- **~Gad ()**  
*Destruktor.*
- bool **getJadowite** () const  
*Pobiera informację, czy zwierzę jest jadowite.*
- int **getDlCiala** () const  
*Pobiera długość ciała zwierzęcia.*
- std::string **getSposobRuchu** () const  
*Pobiera sposób poruszania się zwierzęcia.*
- void **pokazInfo** () const override  
*Wyświetla informacje o gadzie zwierzęciu.*

### Public Member Functions inherited from [Zwierze](#)

- **Zwierze ()**  
*Konstruktor domyślny.*
- **[Zwierze](#) (const std::string &, const std::string &, const std::string &, const std::string &, const int &, const std::string &)**  
*Konstruktor inicjalizujący dane zwierzęcia.*
- **~Zwierze ()**  
*Destruktor.*
- **std::string [getImie](#) () const**  
*Metoda do pobierania imienia zwierzęcia.*
- **std::string [getGatunek](#) () const**  
*Metoda do pobierania gatunku zwierzęcia.*
- **std::string [getPochodzenie](#) () const**  
*Metoda do pobierania pochodzenia zwierzęcia.*
- **std::string [getDieta](#) () const**  
*Metoda do pobierania diety zwierzęcia.*
- **int [getDlZycia](#) () const**  
*Metoda do pobierania długości życia zwierzęcia.*
- **std::string [getSezon](#) () const**  
*Metoda do pobierania sezonu, w którym występuje zwierzę.*
- **virtual void [pokazInfo](#) () const =0**  
*Metoda wirtualna do wyświetlania informacji o zwierzęciu. Metoda wymaga implementacji w klasach pochodnych.*

### Additional Inherited Members

#### Protected Attributes inherited from [Zwierze](#)

- std::string **imie**
- std::string **gatunek**
- std::string **pochodzenie**
- std::string **dieta**
- int **dl\_zycia**
- std::string **sezon**

#### 4.3.1 Detailed Description

Reprezentuje gadzi zwierzę.

Dziedziczy po klasie [Zwierze](#).

#### 4.3.2 Constructor & Destructor Documentation

#### 4.3.2.1 Gad()

```
Gad::Gad (
    const std::string & i,
    const std::string & g,
    const std::string & p,
    const std::string & d,
    const int & dl,
    const std::string & s,
    const bool & j,
    const std::string & sr,
    const int & dlu )
```

Konstruktor inicjalizujący właściwości gada.

##### Parameters

<i>imie</i>	Imię gada.
<i>gatunek</i>	Gatunek gada.
<i>pochodzenie</i>	Pochodzenie gada.
<i>dieta</i>	Dieta gada.
<i>dl_zycia</i>	Długość życia gada.
<i>sezon</i>	Sezon, w którym występuje gad.
<i>jadowite</i>	Flaga określająca czy gad jest jadowity.
<i>s_ruchu</i>	Sposób ruchu gada.
<i>dl_ciala</i>	Długość ciała gada.

### 4.3.3 Member Function Documentation

#### 4.3.3.1 getDlCiala()

```
int Gad::getDlCiala ( ) const
```

Pobiera długość ciała zwierzęcia.

##### Returns

Długość ciała zwierzęcia.

#### 4.3.3.2 getJadowite()

```
bool Gad::getJadowite ( ) const
```

Pobiera informację, czy zwierzę jest jadowite.

##### Returns

Wartość true, jeśli zwierzę jest jadowite. Wartość false w przeciwnym razie.

#### 4.3.3.3 getSposobRuchu()

```
std::string Gad::getSposobRuchu ( ) const
```

Pobiera sposób poruszania się zwierzęcia.

##### Returns

Sposób poruszania się zwierzęcia.

#### 4.3.3.4 pokazInfo()

```
void Gad::pokazInfo ( ) const [override], [virtual]
```

Wyświetla informacje o gadzie zwierzęciu.

Implements [Zwierze](#).

The documentation for this class was generated from the following files:

- Gad.h
- Gad.cpp

## 4.4 Interfejs Class Reference

Klasa reprezentująca interfejs aplikacji.

```
#include <Interfejs.h>
```

### Public Member Functions

- **Interfejs ()**  
*Konstruktor domyślny.*
- **~Interfejs ()**  
*Destruktor.*
- int **menu ()**  
*Wyświetla menu główne aplikacji.*
- int **menuUzytkownik ()**  
*Wyświetla menu dla użytkownika.*
- int **menuPersonel ()**  
*Wyświetla menu dla personelu.*
- void **dodajZwierze ()**  
*Dodaje zwierzę.*
- void **usunZwierze (int numer)**  
*Usuwa zwierzę o podanym imieniu.*
- void **wyswietlImionaZwierzat ()**

- Wyświetla imiona wszystkich zwierząt.*
- void **dodajBilet** ()  
*Dodaje bilet.*
- void **usunBilet** (std::string typ)  
*Usuwa bilet o podanym typie.*
- void **dodajEvent** ()  
*Dodaje wydarzenie.*
- void **usunEvent** (std::string nazwa)  
*Usuwa wydarzenie o podanej nazwie.*
- void **wyswietlCennik** ()  
*Wyświetla cennik biletów.*
- void **wyswietlWydarzenia** ()  
*Wyświetla wszystkie wydarzenia.*
- void **kupBiletPrzezInternet** ()  
*Kupuje bilet przez internet.*
- void **zapiszBilet** (const **Zwiedzajacy** &zwiedzajacy, const **Bilet** &bilet)  
*Zapisuje informacje o zakupionym bilecie do pliku.*
- void **wyswietlKupioneBilety** ()  
*Wyświetla informacje o wszystkich zakupionych biletach.*
- void **wyczyscPlikBiletow** ()  
*Wyczyść zawartość pliku z biletami.*
- void **exitu** ()  
*Obsługuje wyjście z aplikacji dla użytkownika.*
- void **exitp** ()  
*Obsługuje wyjście z aplikacji dla personelu.*

#### 4.4.1 Detailed Description

Klasa reprezentująca interfejs aplikacji.

Zarządza zwierzętami, biletami, wydarzeniami i obsługą użytkownika.

#### 4.4.2 Member Function Documentation

##### 4.4.2.1 menu()

```
int Interfejs::menu ( )
```

Wyświetla menu główne aplikacji.

##### Returns

Wybrana opcja menu.

#### 4.4.2.2 menuPersonel()

```
int Interfejs::menuPersonel ( )
```

Wyświetla menu dla personelu.

##### Returns

Wybrana opcja menu personelu.

#### 4.4.2.3 menuUzytkownik()

```
int Interfejs::menuUzytkownik ( )
```

Wyświetla menu dla użytkownika.

##### Returns

Wybrana opcja menu użytkownika.

#### 4.4.2.4 usunBilet()

```
void Interfejs::usunBilet (
    std::string typ )
```

Usuwa bilet o podanym typie.

##### Parameters

<i>typ</i>	Typ biletu.
------------	-------------

#### 4.4.2.5 usunEvent()

```
void Interfejs::usunEvent (
    std::string nazwa )
```

Usuwa wydarzenie o podanej nazwie.

##### Parameters

<i>nazwa</i>	Nazwa wydarzenia.
--------------	-------------------

#### 4.4.2.6 usunZwierze()

```
void Interfejs::usunZwierze (
    int numer )
```

Usuwa zwierzę o podanym imieniu.

##### Parameters

<i>numer</i>	Numer zwierzęcia z listy.
--------------	---------------------------

#### 4.4.2.7 zapiszBilet()

```
void Interfejs::zapiszBilet (
    const Zwiedzajacy & zwiedzajacy,
    const Bilet & bilet )
```

Zapisuje informacje o zakupionym bilecie do pliku.

##### Parameters

<i>zwiedzajacy</i>	Informacje o zwiedzającym.
<i>bilet</i>	Zakupiony bilet.

The documentation for this class was generated from the following files:

- Interfejs.h
- Interfejs.cpp

## 4.5 Personel Class Reference

Klasa reprezentująca personel obsługujący aplikację.

```
#include <Personel.h>
```

### Public Member Functions

- **Personel** ()  
*Konstruktor domyślny.*
- **Personel** (const std::string &login, const std::string &haslo)  
*Konstruktor inicjalizujący login i hasło personelu.*
- **~Personel** ()  
*Destruktor.*
- void **autoryzacja** ()  
*Metoda do autoryzacji personelu.*



### 4.5.1 Detailed Description

Klasa reprezentująca personel obsługujący aplikację.

### 4.5.2 Constructor & Destructor Documentation

#### 4.5.2.1 Personel()

```
Personel::Personel (
    const std::string & login,
    const std::string & haslo )
```

Konstruktor inicjalizujący login i hasło personelu.

#### Parameters

<i>login</i>	Login personelu.
<i>haslo</i>	Hasło personelu.

The documentation for this class was generated from the following files:

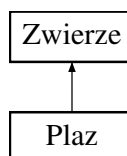
- Personel.h
- Personel.cpp

## 4.6 Plaz Class Reference

Klasa reprezentująca płazy.

```
#include <Plaz.h>
```

Inheritance diagram for Plaz:



### Public Member Functions

- **Plaz ()**  
*Konstruktor domyślny.*
- **Plaz** (const std::string &, const std::string &, const std::string &, const std::string &, const int &, const std::string &, const bool &, const std::string &, const std::string &)

- Konstruktor inicjalizujący właściwości płaza.*
- `~Plaz ()`  
*Destruktor.*
- `bool getWodne () const`  
*Metoda do pobierania informacji o tym, czy płaz jest wodny.*
- `std::string getKolorSkory () const`  
*Metoda do pobierania informacji o kolorze skóry płaza.*
- `std::string getSpRozmnazania () const`  
*Metoda do pobierania informacji o sposobie rozmnażania płaza.*
- `void pokazInfo () const override`  
*Metoda wyświetlająca informacje o płazie.*

### Public Member Functions inherited from `Zwierze`

- `Zwierze ()`  
*Konstruktor domyślny.*
- `Zwierze (const std::string &, const std::string &, const std::string &, const std::string &, const int &, const std::string &)`  
*Konstruktor inicjalizujący dane zwierzęcia.*
- `~Zwierze ()`  
*Destruktor.*
- `std::string getImie () const`  
*Metoda do pobierania imienia zwierzęcia.*
- `std::string getGatunek () const`  
*Metoda do pobierania gatunku zwierzęcia.*
- `std::string getPochodzenie () const`  
*Metoda do pobierania pochodzenia zwierzęcia.*
- `std::string getDieta () const`  
*Metoda do pobierania diety zwierzęcia.*
- `int getDlZycia () const`  
*Metoda do pobierania długości życia zwierzęcia.*
- `std::string getSezon () const`  
*Metoda do pobierania sezonu, w którym występuje zwierzę.*
- `virtual void pokazInfo () const =0`  
*Metoda wirtualna do wyświetlania informacji o zwierzęciu. Metoda wymaga implementacji w klasach pochodnych.*

### Additional Inherited Members

#### Protected Attributes inherited from `Zwierze`

- `std::string imie`
- `std::string gatunek`
- `std::string pochodzenie`
- `std::string dieta`
- `int dl_zycia`
- `std::string sezon`

#### 4.6.1 Detailed Description

Klasa reprezentująca płazy.

## 4.6.2 Constructor & Destructor Documentation

### 4.6.2.1 Plaz()

```
Plaz::Plaz (
    const std::string & i,
    const std::string & g,
    const std::string & p,
    const std::string & d,
    const int & dl,
    const std::string & s,
    const bool & w,
    const std::string & ks,
    const std::string & sr )
```

Konstruktor inicjalizujący właściwości płaza.

#### Parameters

<i>imie</i>	Imię płaza.
<i>gatunek</i>	Gatunek płaza.
<i>pochozenie</i>	Pochodzenie płaza.
<i>dieta</i>	Dieta płaza.
<i>dl_zycia</i>	Długość życia płaza.
<i>sezon</i>	Sezon, w którym występuje płaz.
<i>kolor_skory</i>	Kolor skóry płaza.
<i>wodne</i>	Flaga określająca, czy płaz jest wodny.
<i>sp_rozmnazania</i>	Sposób rozmnażania płaza.

## 4.6.3 Member Function Documentation

### 4.6.3.1 getKolorSkory()

```
std::string Plaz::getKolorSkory ( ) const
```

Metoda do pobierania informacji o kolorze skóry płaza.

#### Returns

Kolor skóry płaza.

#### 4.6.3.2 getSpRozmnazania()

```
std::string Plaz::getSpRozmnazania ( ) const
```

Metoda do pobierania informacji o sposobie rozmnażania płaza.

##### Returns

Sposób rozmnażania płaza.

#### 4.6.3.3 getWodne()

```
bool Plaz::getWodne ( ) const
```

Metoda do pobierania informacji o tym, czy płaz jest wodny.

##### Returns

Wartość logiczna określająca, czy płaz jest wodny.

#### 4.6.3.4 pokazInfo()

```
void Plaz::pokazInfo ( ) const [override], [virtual]
```

Metoda wyświetlająca informacje o płazie.

Implements [Zwierze](#).

The documentation for this class was generated from the following files:

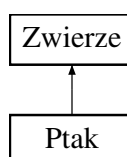
- Plaz.h
- Plaz.cpp

## 4.7 Ptak Class Reference

Klasa reprezentująca ptaki.

```
#include <Ptak.h>
```

Inheritance diagram for Ptak:



## Public Member Functions

- **Ptak** ()  
*Konstruktor domyślny.*
- **Ptak** (const std::string &i, const std::string &g, const std::string &p, const std::string &d, const int &dl, const std::string &s, const bool &latajace, const double &rozpietosc\_skrzydel, const std::string &rodzaj\_gniazda, const bool &migracje)  
*Konstruktor inicjalizujący właściwości ptaka.*
- **~Ptak** ()  
*Destruktor.*
- bool **getLatajacy** () const  
*Metoda do pobierania informacji o tym, czy ptak jest latający.*
- double **getRozpietoscSkrzydel** () const  
*Metoda do pobierania informacji o rozpiętości skrzydeł ptaka.*
- std::string **getRodzajGniazda** () const  
*Metoda do pobierania informacji o rodzaju gniazda, w którym ptak się gnieździ.*
- bool **getMigracje** () const  
*Metoda do pobierania informacji o migracjach ptaka.*
- void **pokazInfo** () const override  
*Metoda wyświetlająca informacje o ptaku.*

## Public Member Functions inherited from **Zwierze**

- **Zwierze** ()  
*Konstruktor domyślny.*
- **Zwierze** (const std::string &, const std::string &, const std::string &, const std::string &, const int &, const std::string &)  
*Konstruktor inicjalizujący dane zwierzęcia.*
- **~Zwierze** ()  
*Destruktor.*
- std::string **getImie** () const  
*Metoda do pobierania imienia zwierzęcia.*
- std::string **getGatunek** () const  
*Metoda do pobierania gatunku zwierzęcia.*
- std::string **getPochodzenie** () const  
*Metoda do pobierania pochodzenia zwierzęcia.*
- std::string **getDieta** () const  
*Metoda do pobierania diety zwierzęcia.*
- int **getDlZycia** () const  
*Metoda do pobierania długości życia zwierzęcia.*
- std::string **getSezon** () const  
*Metoda do pobierania sezonu, w którym występuje zwierzę.*
- virtual void **pokazInfo** () const =0  
*Metoda wirtualna do wyświetlania informacji o zwierzęciu. Metoda wymaga implementacji w klasach pochodnych.*

## Additional Inherited Members

### Protected Attributes inherited from **Zwierze**

- std::string **imie**
- std::string **gatunek**
- std::string **pochodzenie**
- std::string **dieta**
- int **dl\_zycia**
- std::string **sezon**

### 4.7.1 Detailed Description

Klasa reprezentująca ptaki.

### 4.7.2 Constructor & Destructor Documentation

#### 4.7.2.1 Ptak()

```
Ptak::Ptak (
    const std::string & i,
    const std::string & g,
    const std::string & p,
    const std::string & d,
    const int & dl,
    const std::string & s,
    const bool & latajace,
    const double & rozpietosc_skrzydel,
    const std::string & rodzaj_gniazda,
    const bool & migracje )
```

Konstruktor inicjalizujący właściwości ptaka.

#### Parameters

<i>imie</i>	Imię ptaka.
<i>gatunek</i>	Gatunek ptaka.
<i>pochodzenie</i>	Pochodzenie ptaka.
<i>dieta</i>	Dieta ptaka.
<i>dl_zycia</i>	Długość życia ptaka.
<i>sezon</i>	Sezon, w którym występuje ptak.
<i>latajace</i>	Flaga określająca, czy ptak jest latający.
<i>rozpietosc_skrzydel</i>	Rozpiętość skrzydeł ptaka.
<i>rodzaj_gniazda</i>	Rodzaj gniazda, w którym ptak się gnieździ.
<i>migracje</i>	Flaga określająca, czy ptak wykonuje migracje.

### 4.7.3 Member Function Documentation

#### 4.7.3.1 getLatajacy()

```
bool Ptak::getLatajacy ( ) const
```

Metoda do pobierania informacji o tym, czy ptak jest latający.

**Returns**

Wartość logiczna określająca, czy ptak jest latający.

**4.7.3.2 getMigracje()**

```
bool Ptak::getMigracje ( ) const
```

Metoda do pobierania informacji o migracjach ptaka.

**Returns**

Wartość logiczna określająca, czy ptak wykonuje migracje.

**4.7.3.3 getRodzajGniazda()**

```
std::string Ptak::getRodzajGniazda ( ) const
```

Metoda do pobierania informacji o rodzaju gniazda, w którym ptak się gnieździ.

**Returns**

Rodzaj gniazda ptaka.

**4.7.3.4 getRozpietoscSkrzydel()**

```
double Ptak::getRozpietoscSkrzydel ( ) const
```

Metoda do pobierania informacji o rozpiętości skrzydeł ptaka.

**Returns**

Rozpiętość skrzydeł ptaka.

**4.7.3.5 pokazInfo()**

```
void Ptak::pokazInfo ( ) const [override], [virtual]
```

Metoda wyświetlająca informacje o ptaku.

Implements [Zwierze](#).

The documentation for this class was generated from the following files:

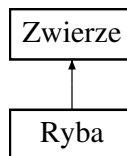
- Ptak.h
- Ptak.cpp

## 4.8 Ryba Class Reference

Klasa reprezentująca ryby.

```
#include <Ryba.h>
```

Inheritance diagram for Ryba:



### Public Member Functions

- **Ryba** ()  
*Konstruktor domyślny.*
- **Ryba** (const std::string &, const std::string &, const std::string &, const std::string &, const int &, const std::string &, const bool &, const double &, const std::string &)  
*Konstruktor inicjalizujący właściwości ryby.*
- **~Ryba** ()  
*Destruktor.*
- bool **getJajorodne** () const  
*Metoda do pobierania informacji o tym, czy ryba jest jajorodna.*
- void **setJajorodne** (bool jajorodne)  
*Metoda do ustawiania informacji o tym, czy ryba jest jajorodna.*
- double **getDlugosc** () const  
*Metoda do pobierania informacji o długości ryby.*
- void **setDlugosc** (double dlugosc)  
*Metoda do ustawiania informacji o długości ryby.*
- std::string **getRodzajSkrzeli** () const  
*Metoda do pobierania informacji o rodzaju skrzeli ryby.*
- void **setRodzajSkrzeli** (const std::string &rodzaj\_skrzeli)  
*Metoda do ustawiania informacji o rodzaju skrzeli ryby.*
- void **pokazInfo** () const override  
*Metoda wyświetlająca informacje o rybie.*

### Public Member Functions inherited from **Zwierze**

- **Zwierze** ()  
*Konstruktor domyślny.*
- **Zwierze** (const std::string &, const std::string &, const std::string &, const std::string &, const int &, const std::string &)  
*Konstruktor inicjalizujący dane zwierzęcia.*
- **~Zwierze** ()  
*Destruktor.*
- std::string **getImie** () const  
*Metoda do pobierania imienia zwierzęcia.*
- std::string **getGatunek** () const



- Metoda do pobierania gatunku zwierzęcia.*
- `std::string getPochodzenie () const`
- Metoda do pobierania pochodzenia zwierzęcia.*
- `std::string getDieta () const`
- Metoda do pobierania diety zwierzęcia.*
- `int getDlZycia () const`
- Metoda do pobierania długości życia zwierzęcia.*
- `std::string getSezon () const`
- Metoda do pobierania sezonu, w którym występuje zwierzę.*
- `virtual void pokazInfo () const =0`
- Metoda wirtualna do wyświetlania informacji o zwierzęciu. Metoda wymaga implementacji w klasach pochodnych.*

## Additional Inherited Members

### Protected Attributes inherited from [Zwierze](#)

- `std::string imie`
- `std::string gatunek`
- `std::string pochodzenie`
- `std::string dieta`
- `int dl_zycia`
- `std::string sezon`

## 4.8.1 Detailed Description

Klasa reprezentująca ryby.

## 4.8.2 Constructor & Destructor Documentation

### 4.8.2.1 Ryba()

```
Ryba::Ryba (
    const std::string & i,
    const std::string & g,
    const std::string & p,
    const std::string & d,
    const int & dl,
    const std::string & s,
    const bool & j,
    const double & dlug,
    const std::string & skrz )
```

Konstruktor inicjalizujący właściwości ryby.

## Parameters

<i>imie</i>	Imię ryby.
<i>gatunek</i>	Gatunek ryby.
<i>pochodzenie</i>	Pochodzenie ryby.
<i>dieta</i>	Dieta ryby.
<i>dl_zycia</i>	Długość życia ryby.
<i>sezon</i>	Sezon, w którym występuje ryba.
<i>jajorodne</i>	Flaga określająca, czy ryba jest jajorodna.
<i>dlugosc</i>	Długość ryby.
<i>rodzaj_skrzeli</i>	Rodzaj skrzeli ryby.

### 4.8.3 Member Function Documentation

#### 4.8.3.1 getDlugosc()

```
double Ryba::getDlugosc ( ) const
```

Metoda do pobierania informacji o długości ryby.

## Returns

Długość ryby.

#### 4.8.3.2 getJajorodne()

```
bool Ryba::getJajorodne ( ) const
```

Metoda do pobierania informacji o tym, czy ryba jest jajorodna.

## Returns

Wartość logiczna określająca, czy ryba jest jajorodna.

#### 4.8.3.3 getRodzajSkrzeli()

```
std::string Ryba::getRodzajSkrzeli ( ) const
```

Metoda do pobierania informacji o rodzaju skrzeli ryby.

## Returns

Rodzaj skrzeli ryby.

#### 4.8.3.4 pokazInfo()

```
void Ryba::pokazInfo ( ) const [override], [virtual]
```

Metoda wyświetlająca informacje o rybie.

Implements [Zwierze](#).

#### 4.8.3.5 setDlugosc()

```
void Ryba::setDlugosc (
    double dlugosc )
```

Metoda do ustawiania informacji o długości ryby.

##### Parameters

<i>dlugosc</i>	Długość ryby.
----------------	---------------

#### 4.8.3.6 setJajorodne()

```
void Ryba::setJajorodne (
    bool jajorodne )
```

Metoda do ustawiania informacji o tym, czy ryba jest jajorodna.

##### Parameters

<i>jajorodne</i>	Wartość logiczna określająca, czy ryba jest jajorodna.
------------------	--

#### 4.8.3.7 setRodzajSkrzeli()

```
void Ryba::setRodzajSkrzeli (
    const std::string & rodzaj_skrzeli )
```

Metoda do ustawiania informacji o rodzaju skrzeli ryby.

##### Parameters

<i>rodzaj_skrzeli</i>	Rodzaj skrzeli ryby.
-----------------------	----------------------

The documentation for this class was generated from the following files:

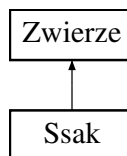
- Ryba.h
- Ryba.cpp

## 4.9 Ssak Class Reference

Klasa reprezentująca ssaki.

```
#include <Ssak.h>
```

Inheritance diagram for Ssak:



### Public Member Functions

- **Ssak** ()  
*Konstruktor domyślny.*
- **Ssak** (const std::string &, const std::string &, const std::string &, const std::string &, const int &, const std::string &, const std::string &, const bool &, const int &, const std::string &)  
*Konstruktor inicjalizujący właściwości ssaka.*
- **~Ssak** ()  
*Destruktor.*
- bool **getZagrozony** () const  
*Metoda do pobierania informacji o tym, czy ssak jest zagrożony.*
- int **getLiczbaPazurow** () const  
*Metoda do pobierania informacji o liczbie pazurów ssaka.*
- std::string **getTrybZycia** () const  
*Metoda do pobierania informacji o trybie życia ssaka.*
- void **pokazInfo** () const override  
*Metoda wyświetlająca informacje o ssaku.*

### Public Member Functions inherited from **Zwierze**

- **Zwierze** ()  
*Konstruktor domyślny.*
- **Zwierze** (const std::string &, const std::string &, const std::string &, const std::string &, const int &, const std::string &)  
*Konstruktor inicjalizujący dane zwierzęcia.*
- **~Zwierze** ()  
*Destruktor.*
- std::string **getImie** () const  
*Metoda do pobierania imienia zwierzęcia.*
- std::string **getGatunek** () const  
*Metoda do pobierania gatunku zwierzęcia.*
- std::string **getPochodzenie** () const

- *Metoda do pobierania pochodzenia zwierzęcia.*  
std::string [getDieta](#) () const
- *Metoda do pobierania diety zwierzęcia.*  
int [getDlZycia](#) () const
- *Metoda do pobierania długości życia zwierzęcia.*  
std::string [getSezon](#) () const
- *Metoda do pobierania sezonu, w którym występuje zwierzę.*  
virtual void [pokazInfo](#) () const =0
- *Metoda wirtualna do wyświetlania informacji o zwierzęciu. Metoda wymaga implementacji w klasach pochodnych.*

## Additional Inherited Members

### Protected Attributes inherited from [Zwierze](#)

- std::string **imie**
- std::string **gatunek**
- std::string **pochodzenie**
- std::string **dieta**
- int **dl\_zycia**
- std::string **sezon**

## 4.9.1 Detailed Description

Klasa reprezentująca ssaki.

## 4.9.2 Constructor & Destructor Documentation

### 4.9.2.1 Ssak()

```
Ssak::Ssak (
    const std::string & i,
    const std::string & g,
    const std::string & p,
    const std::string & d,
    const int & dl,
    const std::string & s,
    const bool & z,
    const int & lp,
    const std::string & tz )
```

Konstruktor inicjalizujący właściwości ssaka.

#### Parameters

<i>imie</i>	Imię ssaka.
<i>gatunek</i>	Gatunek ssaka.
<i>pochodzenie</i>	Pochodzenie ssaka.
<i>dieta</i>	Dieta ssaka.
<i>dl_zycia</i>	Długość życia ssaka.
<i>sezon</i>	Sezon, w którym występuje ssak.
<i>zagrozone</i>	Flaga określająca, czy ssak jest zagrożony.

## 4.9.3 Member Function Documentation

### 4.9.3.1 getLiczbaPazurow()

```
int Ssak::getLiczbaPazurow ( ) const
```

Metoda do pobierania informacji o liczbie pazurów ssaka.

#### Returns

Liczba pazurów ssaka.

### 4.9.3.2 getTrybZycia()

```
std::string Ssak::getTrybZycia ( ) const
```

Metoda do pobierania informacji o trybie życia ssaka.

#### Returns

Tryb życia ssaka.

### 4.9.3.3 getZagrozony()

```
bool Ssak::getZagrozony ( ) const
```

Metoda do pobierania informacji o tym, czy ssak jest zagrożony.

#### Returns

Wartość logiczna określająca, czy ssak jest zagrożony.

### 4.9.3.4 pokazInfo()

```
void Ssak::pokazInfo ( ) const [override], [virtual]
```

Metoda wyświetlająca informacje o ssaku.

Implements [Zwierze](#).

The documentation for this class was generated from the following files:

- Ssak.h
- Ssak.cpp

## 4.10 Zoo Class Reference

### Public Member Functions

- void **dodajZwierze** ()
- void **usunZwierze** (std::string imie)
- void **wyszukajZwierze** (std::string &imie)
- void **dodajBilet** ()
- void **usunBilet** (std::string typ)
- void **dodajEvent** ()

The documentation for this class was generated from the following files:

- Zoo.h
- Zoo.cpp

## 4.11 Zwiedzajacy Class Reference

Klasa reprezentująca zwiedzającego.

```
#include <Zwiedzajacy.h>
```

### Public Member Functions

- [Zwiedzajacy](#) (const std::string &, const std::string &, const std::string &)  
*Konstruktor inicjalizujący dane zwiedzającego.*
- [~Zwiedzajacy](#) ()  
*Destruktor.*
- std::string [getImie](#) () const  
*Metoda do pobierania imienia zwiedzającego.*
- std::string [getNazwisko](#) () const  
*Metoda do pobierania nazwiska zwiedzającego.*
- std::string [getNrTel](#) () const  
*Metoda do pobierania numeru telefonu zwiedzającego.*

#### 4.11.1 Detailed Description

Klasa reprezentująca zwiedzającego.

#### 4.11.2 Constructor & Destructor Documentation

##### 4.11.2.1 Zwiedzajacy()

```
Zwiedzajacy::Zwiedzajacy (  
    const std::string & im,  
    const std::string & naz,  
    const std::string & nr_tel )
```

Konstruktor inicjalizujący dane zwiedzającego.

**Parameters**

<i>imie</i>	Imię zwiedzającego.
<i>nazwisko</i>	Nazwisko zwiedzającego.
<i>numer</i>	Numer telefonu zwiedzającego.

### 4.11.3 Member Function Documentation

#### 4.11.3.1 `getImie()`

```
std::string Zwiedzajacy::getImie ( ) const
```

Metoda do pobierania imienia zwiedzającego.

**Returns**

Imię zwiedzającego.

#### 4.11.3.2 `getNazwisko()`

```
std::string Zwiedzajacy::getNazwisko ( ) const
```

Metoda do pobierania nazwiska zwiedzającego.

**Returns**

Nazwisko zwiedzającego.

#### 4.11.3.3 `getNrTel()`

```
std::string Zwiedzajacy::getNrTel ( ) const
```

Metoda do pobierania numeru telefonu zwiedzającego.

**Returns**

Numer telefonu zwiedzającego.

The documentation for this class was generated from the following files:

- `Zwiedzajacy.h`
- `Zwiedzajacy.cpp`

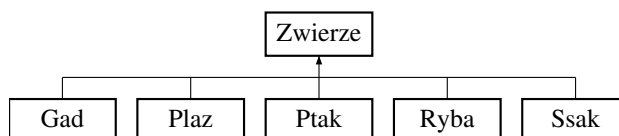


## 4.12 Zwierze Class Reference

Klasa przechowująca informacje o zwierzętach.

```
#include <Zwierze.h>
```

Inheritance diagram for Zwierze:



### Public Member Functions

- **Zwierze ()**  
*Konstruktor domyślny.*
- **Zwierze (const std::string &, const std::string &, const std::string &, const std::string &, const int &, const std::string &)**  
*Konstruktor inicjalizujący dane zwierzęcia.*
- **~Zwierze ()**  
*Destruktor.*
- **std::string getImie () const**  
*Metoda do pobierania imienia zwierzęcia.*
- **std::string getGatunek () const**  
*Metoda do pobierania gatunku zwierzęcia.*
- **std::string getPochodzenie () const**  
*Metoda do pobierania pochodzenia zwierzęcia.*
- **std::string getDieta () const**  
*Metoda do pobierania diety zwierzęcia.*
- **int getDlZycia () const**  
*Metoda do pobierania długości życia zwierzęcia.*
- **std::string getSezon () const**  
*Metoda do pobierania sezonu, w którym występuje zwierzę.*
- **virtual void pokazInfo () const =0**  
*Metoda wirtualna do wyświetlania informacji o zwierzęciu. Metoda wymaga implementacji w klasach pochodnych.*

### Protected Attributes

- **std::string imie**
- **std::string gatunek**
- **std::string pochodzenie**
- **std::string dieta**
- **int dl\_zycia**
- **std::string sezon**

#### 4.12.1 Detailed Description

Klasa przechowująca informacje o zwierzętach.

## 4.12.2 Constructor & Destructor Documentation

### 4.12.2.1 Zwierze()

```
Zwierze::Zwierze (
    const std::string & i,
    const std::string & g,
    const std::string & p,
    const std::string & d,
    const int & dl,
    const std::string & s )
```

Konstruktor inicjalizujący dane zwierzęcia.

#### Parameters

<i>imie</i>	Imię zwierzęcia.
<i>gatunek</i>	Gatunek zwierzęcia.
<i>pochodzenie</i>	Pochodzenie zwierzęcia.
<i>dieta</i>	Dieta zwierzęcia.
<i>dl_zycia</i>	Długość życia zwierzęcia.
<i>sezon</i>	Sezon, w którym występuje zwierzę.

## 4.12.3 Member Function Documentation

### 4.12.3.1 getDieta()

```
std::string Zwierze::getDieta ( ) const
```

Metoda do pobierania diety zwierzęcia.

#### Returns

Dieta zwierzęcia.

### 4.12.3.2 getDlZycia()

```
int Zwierze::getDlZycia ( ) const
```

Metoda do pobierania długości życia zwierzęcia.

#### Returns

Długość życia zwierzęcia.

#### 4.12.3.3 getGatunek()

```
std::string Zwierze::getGatunek ( ) const
```

Metoda do pobierania gatunku zwierzęcia.

##### Returns

Gatunek zwierzęcia.

#### 4.12.3.4 getImie()

```
std::string Zwierze::getImie ( ) const
```

Metoda do pobierania imienia zwierzęcia.

##### Returns

Imię zwierzęcia.

#### 4.12.3.5 getPochodzenie()

```
std::string Zwierze::getPochodzenie ( ) const
```

Metoda do pobierania pochodzenia zwierzęcia.

##### Returns

Pochodzenie zwierzęcia.

#### 4.12.3.6 getSezon()

```
std::string Zwierze::getSezon ( ) const
```

Metoda do pobierania sezonu, w którym występuje zwierzę.

##### Returns

Sezon występowania zwierzęcia.

#### 4.12.3.7 pokazInfo()

```
void Zwierze::pokazInfo ( ) const [pure virtual]
```

Metoda wirtualna do wyświetlania informacji o zwierzęciu. Metoda wymaga implementacji w klasach pochodnych.

Implemented in [Gad](#), [Plaz](#), [Ptak](#), [Ryba](#), and [Ssak](#).

The documentation for this class was generated from the following files:

- [Zwierze.h](#)
- [Zwierze.cpp](#)



## Chapter 5

# File Documentation

### 5.1 Bilet.h

```
00001 #pragma once
00002
00003 #include <string>
00004 #include <fstream>
00005 #include <vector>
00006 #include <sstream>
00007 #include <iostream>
00008
00013 class Bilet {
00014     std::string typ;
00015     double cena;
00016 public:
00020     Bilet();
00021
00027     Bilet(const std::string& typ, const double& cena);
00028
00032     ~Bilet();
00033
00038     std::string getTyp() const;
00039
00044     double getCena() const;
00045
00049     virtual void pokazInfo() const;
00050
00056     static std::vector<Bilet> odczytajWszystkieZPliku(const std::string& nazwapliku);
00057 };
```

### 5.2 Event.h

```
00001 #pragma once
00002
00003 #include <iostream>
00004 #include <string>
00005 #include <vector>
00006 #include <fstream>
00007 #include <sstream>
00008
00013 class Event {
00014     std::string nazwa;
00015     std::string czas;
00016     std::string data;
00017 public:
00021     Event();
00022
00029     Event(const std::string& nazwa, const std::string& czas, const std::string& data);
00030
00034     ~Event();
00035
00040     std::string getNazwa() const;
00041
00046     std::string getCzas() const;
00047
00052     std::string getData() const;
00053 }
```

```

00059     static std::vector<Event> odczytajWszystkieZPliku(const std::string& nazwapliku);
00060
00064     void godzinyOtwarcia();
00065
00076     friend std::ostream& operator<<(std::ostream& os, const Event& event);
00077 };

```

## 5.3 Gad.h

```

00001 #pragma once
00002 #include "Zwierze.h"
00003
00009 class Gad : public Zwierze {
00010     bool jadowite;
00011     int dl_ciala;
00012     std::string s_ruchu;
00013
00014 public:
00018     Gad();
00019
00032     Gad(const std::string&, const std::string&, const std::string&, const std::string&, const int&,
const std::string&, const bool&, const std::string&, const int&);
00033
00037     ~Gad();
00038
00043     bool getJadowite() const;
00044
00049     int getDlCiala() const;
00050
00055     std::string getSposobRuchu() const;
00056
00060     void pokazInfo() const override;
00061 };

```

## 5.4 Interfejs.h

```

00001 #pragma once
00002 #ifndef INTERFEJS_H
00003 #define INTERFEJS_H
00004
00005 #include <string>
00006 #include <vector>
00007 #include <fstream>
00008 #include <sstream>
00009 #include <algorithm>
00010
00011 #include "Zwierze.h"
00012 #include "Bilet.h"
00013 #include "Event.h"
00014 #include "Plaz.h"
00015 #include "Ssak.h"
00016 #include "Gad.h"
00017 #include "Ptak.h"
00018 #include "Ryba.h"
00019 #include "Personel.h"
00020 #include "Zwiedzajacy.h"
00021
00027 class Interfejs {
00028     std::vector<Zwierze*> zwierzeta;
00029     std::vector<Bilet*> bilety;
00030     std::vector<Event*> wydarzenia;
00031     std::vector<std::string> rodzajeBiletow;
00032     std::vector<double> cenyBiletow;
00033 public:
00037     Interfejs();
00038
00042     ~Interfejs();
00043
00048     int menu();
00049
00054     int menuUzytkownik();
00055
00060     int menuPersonel();
00061
00065     void dodajZwierze();
00066
00071     void usunZwierze(int numer);
00072
00076     void wyswietlImionaZwierzat();

```

```

00077
00081     void dodajBilet();
00082
00087     void usunBilet(std::string typ);
00088
00092     void dodajEvent();
00093
00098     void usunEvent(std::string nazwa);
00099
00103     void wyswietlCennik();
00104
00108     void wyswietlWydarzenia();
00109
00113     void kupBiletPrzezInternet();
00114
00120     void zapiszBilet(const Zwiedzajacy& zwiedzajacy, const Bilet& bilet);
00121
00125     void wyswietlKupioneBilety();
00126
00130     void wyczyscPlikBiletow();
00131
00135     void exitu();
00136
00140     void exitp();
00141 };
00142
00143 #endif

```

## 5.5 Personel.h

```

00001 #ifndef PERSONEL_H
00002 #define PERSONEL_H
00003
00004 #include "Interfejs.h"
00005 #include <string>
00006
00011 class Personel {
00012     std::string login;
00013     std::string haslo;
00014
00015 public:
00019     Personel();
00020
00026     Personel(const std::string& login, const std::string& haslo);
00027
00031     ~Personel();
00032
00036     void autoryzacja();
00037 };
00038
00039 #endif // PERSONEL_H

```

## 5.6 Plaz.h

```

00001 #pragma once
00002 #include "Zwierze.h"
00003
00008 class Plaz : public Zwierze {
00009     bool wodne;
00010     std::string kolor_skory;
00011     std::string sp_rozmnazania;
00012
00013 public:
00017     Plaz();
00018
00031     Plaz(const std::string&, const std::string&, const std::string&, const std::string&, const int&,
00032         const std::string&, const bool&, const std::string&, const std::string&);
00033
00036     ~Plaz();
00037
00042     bool getWodne() const;
00043
00048     std::string getKolorSkory() const;
00049
00054     std::string getSpRozmnazania() const;
00055
00059     void pokazInfo() const override;
00060 };

```

## 5.7 Ptak.h

```
00001 #pragma once
00002 #include "Zwierze.h"
00003
00008 class Ptak : public Zwierze {
00009     bool latajace;
00010     double rozpietosc_skrzydel;
00011     std::string rodzaj_gniazda;
00012     bool migracje;
00013
00014 public:
00018     Ptak();
00019
00033     Ptak(const std::string& i, const std::string& g, const std::string& p, const std::string& d, const
int& dl, const std::string& s, const bool& latajace, const double& rozpietosc_skrzydel, const
std::string& rodzaj_gniazda, const bool& migracje);
00034
00038     ~Ptak();
00039
00044     bool getLatajacy() const;
00045
00050     double getRozpietoscSkrzydel() const;
00051
00056     std::string getRodzajGniazda() const;
00057
00062     bool getMigracje() const;
00063
00067     void pokazInfo() const override;
00068 };
```

## 5.8 Ryba.h

```
00001 #pragma once
00002 #include <string>
00003 #include "Zwierze.h"
00004
00009 class Ryba : public Zwierze {
00010     bool jajorodne;
00011     double dlugosc;
00012     std::string rodzaj_skrzeli;
00013
00014 public:
00018     Ryba();
00019
00032     Ryba(const std::string&, const std::string&, const std::string&, const std::string&, const int&,
const std::string&, const bool&, const double&, const std::string&);
00033
00037     ~Ryba();
00038
00043     bool getJajorodne() const;
00044
00049     void setJajorodne(bool jajorodne);
00050
00055     double getDlugosc() const;
00056
00061     void setDlugosc(double dlugosc);
00062
00067     std::string getRodzajSkrzeli() const;
00068
00073     void setRodzajSkrzeli(const std::string& rodzaj_skrzeli);
00074
00078     void pokazInfo() const override;
00079 };
```

## 5.9 Ssak.h

```
00001 #pragma once
00002 #include "Zwierze.h"
00003
00008 class Ssak : public Zwierze {
00009     bool zagrozone;
00010     int liczba_pazurow;
00011     std::string tryb_zycia;
00012
00013 public:
00017     Ssak();
00018
```



```

00031     Ssak(const std::string&, const std::string&, const std::string&, const std::string&, const int&,
const std::string&, const bool&, const int&, const std::string&);
00032
00036     ~Ssak();
00037
00042     bool getZagrozony() const;
00043
00048     int getLiczbaPazurow() const;
00049
00054     std::string getTrybZycia() const;
00055
00059     void pokazInfo() const override;
00060 };

```

## 5.10 Zoo.h

```

00001 // #include <iostream>
00002 // #include <string>
00003 // #include <vector>
00004 // #include "Zwierze.h"
00005 // #include "Bilet.h"
00006 // #include "Event.h"
00007 //
00008 // class Zoo {
00009 // public:
00010 //     Zoo();
00011 //     Zoo(const std::string& nazwa, const std::string& adres);
00012 //
00013 //     void dodajZwierze(const Zwierze& zwierze);
00014 //     void usunZwierze(const std::string& imie);
00015 //     std::vector<Zwierze> getZwierzeta() const;
00016 //     std::vector<Zwierze> szukajZwierzatPoImieniu(const std::string& imie) const;
00017 //
00018 //     void dodajBilet(const Bilet& bilet);
00019 //     void usunBilet(const std::string& typ);
00020 //     std::vector<Bilet> getBilety() const;
00021 //
00022 //     void dodajEvent(const Event& event);
00023 //     void usunEvent(const std::string& nazwa);
00024 //     std::vector<Event> getEvents() const;
00025 //     std::vector<Event> szukajEventuPoNazwie(const std::string& nazwa);
00026 //
00027 //     void pokazGodzinyOtwarcia() const;
00028 //     void pokazCaleInfo() const;
00029 // private:
00030 //     std::string nazwa;
00031 //     std::string adres;
00032 //
00033 //     std::vector<Zwierze> zwierzeta;
00034 //     std::vector<Bilet> bilety;
00035 //     std::vector<Event> eventy;
00036 //     std::vector<std::string> godzinyotwarcia;
00037 // };
00038
00039 #ifndef ZOO_H
00040 #define ZOO_H
00041
00042 #include <string>
00043 #include <vector>
00044
00045 #include "Zwierze.h"
00046 #include "Bilet.h"
00047 #include "Event.h"
00048 #include "Plaz.h"
00049 #include "Ryba.h"
00050 #include "Ssak.h"
00051 #include "Gad.h"
00052 #include "Ptak.h"
00053 #include "Ryba.h"
00054
00055 class Zoo {
00056 public:
00057     Zoo();
00058     ~Zoo();
00059     void dodajZwierze();
00060     void usunZwierze(std::string imie);
00061     void wyszukajZwierze(std::string& imie);
00062     //void wyszukajZwierze(const std::string&) const;
00063     void dodajBilet();
00064     void usunBilet(std::string typ);
00065     void dodajEvent();
00066     //void usunEvent(std::string nazwa);
00067     // void wyszukajEvent(std::string nazwa);

```

```

00068 private:
00069     std::vector<Zwierze*> zwierzeta;
00070     std::vector<Bilet*> bilety;
00071     std::vector<Event*> wydarzenia;
00072 };
00073
00074 #endif

```

## 5.11 Zwiedzajacy.h

```

00001 #include <iostream>
00002 #include <vector>
00003 #include <string>
00004
00009 class Zwiedzajacy {
00010     std::string imie;
00011     std::string nazwisko;
00012     std::string numer;
00013     bool czyBiletKupiony;
00014     std::vector<std::string> listaZwierzat;
00015
00016 public:
00023     Zwiedzajacy(const std::string&, const std::string&, const std::string&);
00024
00028     ~Zwiedzajacy();
00029
00034     std::string getImie() const;
00035
00040     std::string getNazwisko() const;
00041
00046     std::string getNrTel() const;
00047 };

```

## 5.12 Zwierze.h

```

00001 #pragma once
00002 #include <iostream>
00003 #include <fstream>
00004 #include <cstdlib>
00005 #include <string>
00006 #include <vector>
00007
00012 class Zwierze {
00013 protected:
00014     std::string imie;
00015     std::string gatunek;
00016     std::string pochodzenie;
00017     std::string dieta;
00018     int dl_zycia;
00019     std::string sezon;
00020
00021 public:
00025     Zwierze();
00026
00036     Zwierze(const std::string&, const std::string&, const std::string&, const std::string&, const
int&, const std::string&);
00037
00041     ~Zwierze();
00042
00047     std::string getImie() const;
00048
00053     std::string getGatunek() const;
00054
00059     std::string getPochodzenie() const;
00060
00065     std::string getDieta() const;
00066
00071     int getDlZycia() const;
00072
00077     std::string getSezon() const;
00078
00083     virtual void pokazInfo() const = 0;
00084 };

```

# Index

- Bilet, [7](#)
  - Bilet, [8](#)
  - getCena, [8](#)
  - getTyp, [8](#)
  - odczytajWszystkieZPliku, [8](#)
- Event, [9](#)
  - Event, [10](#)
  - getCzas, [10](#)
  - getData, [10](#)
  - getNazwa, [10](#)
  - odczytajWszystkieZPliku, [11](#)
  - operator<<, [11](#)
- Gad, [12](#)
  - Gad, [13](#)
  - getDICiala, [14](#)
  - getJadowite, [14](#)
  - getSposobRuchu, [14](#)
  - pokazInfo, [15](#)
- getCena
  - Bilet, [8](#)
- getCzas
  - Event, [10](#)
- getData
  - Event, [10](#)
- getDieta
  - Zwierze, [36](#)
- getDICiala
  - Gad, [14](#)
- getDlugosc
  - Ryba, [28](#)
- getDIZycia
  - Zwierze, [36](#)
- getGatunek
  - Zwierze, [36](#)
- getImie
  - Zwiedzajacy, [34](#)
  - Zwierze, [37](#)
- getJadowite
  - Gad, [14](#)
- getJajorodne
  - Ryba, [28](#)
- getKolorSkory
  - Plaz, [21](#)
- getLatajacy
  - Ptak, [24](#)
- getLiczbaPazurow
  - Ssak, [32](#)
- getMigracje
  - Ptak, [25](#)
- getNazwa
  - Event, [10](#)
- getNazwisko
  - Zwiedzajacy, [34](#)
- getNrTel
  - Zwiedzajacy, [34](#)
- getPochodzenie
  - Zwierze, [37](#)
- getRodzajGniazda
  - Ptak, [25](#)
- getRodzajSkrzeli
  - Ryba, [28](#)
- getRozpietoscSkrzydel
  - Ptak, [25](#)
- getSezon
  - Zwierze, [37](#)
- getSposobRuchu
  - Gad, [14](#)
- getSpRozmnazania
  - Plaz, [21](#)
- getTrybZycia
  - Ssak, [32](#)
- getTyp
  - Bilet, [8](#)
- getWodne
  - Plaz, [22](#)
- getZagrozony
  - Ssak, [32](#)
- Interfejs, [15](#)
  - menu, [16](#)
  - menuPersonel, [16](#)
  - menuUzytkownik, [17](#)
  - usunBilet, [17](#)
  - usunEvent, [17](#)
  - usunZwierze, [18](#)
  - zapiszBilet, [18](#)
- menu
  - Interfejs, [16](#)
- menuPersonel
  - Interfejs, [16](#)
- menuUzytkownik
  - Interfejs, [17](#)
- odczytajWszystkieZPliku
  - Bilet, [8](#)
  - Event, [11](#)
- operator<<

Event, 11

Personel, 18

    Personel, 19

Plaz, 19

    getKolorSkory, 21

    getSpRozmnazania, 21

    getWodne, 22

    Plaz, 21

    pokazInfo, 22

pokazInfo

    Gad, 15

    Plaz, 22

    Ptak, 25

    Ryba, 28

    Ssak, 32

    Zwierze, 37

Ptak, 22

    getLatajacy, 24

    getMigracje, 25

    getRodzajGniazda, 25

    getRozpietoscSkrzydel, 25

    pokazInfo, 25

    Ptak, 24

Ryba, 26

    getDlugosc, 28

    getJajorodne, 28

    getRodzajSkrzeli, 28

    pokazInfo, 28

    Ryba, 27

    setDlugosc, 29

    setJajorodne, 29

    setRodzajSkrzeli, 29

setDlugosc

    Ryba, 29

setJajorodne

    Ryba, 29

setRodzajSkrzeli

    Ryba, 29

Ssak, 30

    getLiczbaPazurow, 32

    getTrybZycia, 32

    getZagrozony, 32

    pokazInfo, 32

    Ssak, 31

usunBilet

    Interfejs, 17

usunEvent

    Interfejs, 17

usunZwierze

    Interfejs, 18

zapiszBilet

    Interfejs, 18

Zoo, 33

Zwiedzajacy, 33

    getImie, 34

    getNazwisko, 34

    getNrTel, 34

    Zwiedzajacy, 33

Zwierze, 35

    getDieta, 36

    getDIZycia, 36

    getGatunek, 36

    getImie, 37

    getPochodzenie, 37

    getSezon, 37

    pokazInfo, 37

    Zwierze, 36