

Politechnika Śląska

Wydział Automatyki, Elektroniki i Informatyki

Programowanie Komputerów

Statki

| autor | prowadzący | rok akademicki | kierunek | rodzaj studiów | semestr | termin laboratorium | sekcja | Termin oddania sprawozdania |
|---------------|----------------------------------|----------------|-------------|----------------|---------|--|--------|-----------------------------|
| Adam Warzecha | mgr inż. Grzegorz Kwiatkowski | 2020/2021 | informatyka | SSI | 3 | czwartek, 11:15-12:45 piątek, 8:15-9:45 | 2.2 | 03.11.2020 r. |

Link do GitHub: <https://github.com/polsl-aei-pk3/195abe22-gr22-repo>

1. Analiza zadania

Program pobiera dane podane naprzemiennie od użytkowników przez cały czas trwania rozgrywki. Podane dane są przetwarzane. Po zakończeniu rozgrywki wyniki są zapisywane do plików tekstowych.

1.1 Struktury danych

Struktury wykorzystane w programie to: Lista jednokierunkowa – dla przechowywania kolejnych elementów. Każdy element symbolizuje jedno pole na mapie.

1.2 Kluczowe metody

Jedną z najbardziej kluczowych metod w programie jest metoda typu void klasy **Mapa** `zapiszMape(wygladMap*& stat)`. Metoda przyjmuje jako parametr oryginał początku listy jednokierunkowej. Następnie zapisuje do listy całą mapę, wraz z statkami gracza. Następną ważną metodą jest metoda typu void `zapiszDoPlikuGracza1(fstream& plik)`. Metoda klasy **Gracz** przyjmuje jako parametr plik, do którego zostanie zapisana mapa statków, mapa strzałów oraz punktacja. Dla drugiego gracza metoda jest analogiczna.

2. Specyfikacja zewnętrzna

Program uruchomiony jest przez lokalny Debugger w programie Visual Studio. Aby rozpocząć rozgrywkę należy wprowadzić odpowiedni rozmiar.

3. Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem obiektowym oraz posiada jedną dynamiczną strukturę danych. Program opiera się głównie na komunikacji z użytkownikiem. W programie rozdzielono interfejs (komunikację z użytkownikiem) od logiki aplikacji (tworzenie odpowiednio sformatowanych map oraz plików).

4. Testowanie

Program został przetestowany na różnego rodzaju danych wejściowych. Wprowadzanie błędnych statków nie powoduje błędów, jednakże owe statki nie zostaną uwzględnione na mapie. Użytkownik może wtedy rozpocząć rozgrywkę od nowa i podać poprawne dane. Program został sprawdzony pod kątem wycieków pamięci.

5. Instrukcja obsługi

a) Na początku gracze powinni się zapoznać z uproszczoną instrukcją, która zostaje wyświetlona po uruchomieniu programu oraz wprowadzić '1' aby przejść dalej.

```
STATKI
-----
Instrukcja obsługi:
[1] Aby rozpocząć grę należy podać rozmiar mapy: '1'- mapa 4x4, '2'-mapa 6x6, '3'-mapa 10x10
Taki jaki zostanie wybrany rozmiar mapy tyle będzie statków(2,3,4)

[2] Współrzędne statków należy wprowadzać dużymi cyframi. Przykładowa współrzędna X=A, Y=1.
Jeśli zostaną wprowadzone złe współrzędne statki nie zapiszą się poprawnie

[3] Jeżeli współrzędne statków będą się pokrywać lub wychodzić poza zakres mapy,
nie zostanie zapisana prawidłowa ilość statków

[4] Wygrywa gracz, który pierwszy zniszczy wszystkie statki przeciwnika, wyniki są zapisywane do plików tekstowych

Wprowadz '1' aby kontynuować:
```

b) Następnie gracze muszą podać na jak dużej mapie rozgrywka ma się odbywać.

```
Podaj wielkość mapy: '1' - mała mapa, '2' średnia mapa, '3' - duża mapa
Wprowadz: 1
```

c) Na powyższym zdjęciu został wybrany rozmiar 1, czyli rozgrywka będzie się na mapie 4x4 a statkami tutaj będą łódka oraz statek. Pierwszy gracz wprowadza współrzędne dla łódki i następnie dla statku.

```
Podaj poniżej współrzędne statków:
-----
Współrzędna X Łódki: A
Współrzędna Y Łódki: 3
```

```
Podaj poniżej współrzędne Statku:
-----
Współrzędna X Statku: B
Współrzędna Y Statku: 2
Współrzędna X1 Statku: B
Współrzędna Y1 Statku: 3
```

Analogicznie gracz drugi wprowadza swoje statki, pierwszy gracz powinien się odwrócić.

```
Podaj ponizej wspolrzedne statkow:
-----
Wspolrzedna X Lodki: D
Wspolrzedna Y Lodki: 1
```

```
Podaj ponizej wspolrzedne Statku:
-----
Wspolrzedna X Statku: A
Wspolrzedna Y Statku: 1
Wspolrzedna X1 Statku: A
Wspolrzedna Y1 Statku: 2
```

d) Po wprowadzeniu współrzędnych przez graczy rozpoczyna się właściwa rozgrywka, czyli strzały. Zaczyna gracz, który pierwszy wprowadzał statki. Na ekranie zostają wypisane 2 mapy. Pierwsza mapa to mapa statków gracza oraz strzałów przeciwnika. Druga mapa to mapa strzałów gracza.

```
Gracz 1:
Punkty: 0

  1  2  3  4
A [#][#][L][#]
B [#][S][S][#]
C [#][#][#][#]
D [#][#][#][#]

  1  2  3  4
A [#][#][#][#]
B [#][#][#][#]
C [#][#][#][#]
D [#][#][#][#]

Atakuj!

Podaj wspolrzedna X:
```

e) gracz atakuje po przez wprowadzenie współrzędnych na jakie chce oddać strzał.

```
Gracz 1:
Punkty: 0

  1  2  3  4
A [#][#][L][#]
B [#][S][S][#]
C [#][#][#][#]
D [#][#][#][#]

  1  2  3  4
A [#][#][#][#]
B [#][#][#][#]
C [#][#][#][#]
D [#][#][#][#]

Atakuj!

Podaj współrzędna X: A

Podaj współrzędna Y: 2
```

f) Po zatwierdzeniu zostaje wpisany komunikat o powodzeniu trafienia. Należy wprowadzić '1' aby przejść dalej.

```
Gracz 1:
Punkty: 0

  1  2  3  4
A [#][#][L][#]
B [#][S][S][#]
C [#][#][#][#]
D [#][#][#][#]

  1  2  3  4
A [#][#][#][#]
B [#][#][#][#]
C [#][#][#][#]
D [#][#][#][#]

Atakuj!

Podaj współrzędna X: A

Podaj współrzędna Y: 2
BRAWO! TRAFIŁES STATEK PRZECWINIKA!

Wprowadz '1' aby przejść dalej: 1
```

g) Następuje aktualizacja mapy. Aby rozpocząć ruch drugiego gracza należy wprowadzić i zatwierdzić '1'.

```
Gracz 1:
Punkty: 5

  1  2  3  4
A [#][#][L][#]
B [#][S][S][#]
C [#][#][#][#]
D [#][#][#][#]

  1  2  3  4
A [#][X][#][#]
B [#][#][#][#]
C [#][#][#][#]
D [#][#][#][#]

Wprowadz '1' aby przejść dalej: 1
```

h) Wygląd map dla gracza drugiego. Kawałek statku został zatopiony. Gracz drugi oddaje strzał analogicznie do gracza pierwszego. Rozgrywka toczy się, aż do zatopienia wszystkich statków.

```
Gracz 2:
Punkty: 0

  1  2  3  4
A [S][X][#][#]
B [#][#][#][#]
C [#][#][#][#]
D [L][#][#][#]

  1  2  3  4
A [#][#][#][#]
B [#][#][#][#]
C [#][#][#][#]
D [#][#][#][#]

Atakuj!

Podaj współrzędna X:
```

i) Gdy jeden z graczy zatopi wszystkie statki przeciwnika zostaje wypisany komunikat (w tym przypadku gracz 1).

```
Gracz1 wygrywa!  
Press any key to continue . . .
```

j) Wyniki zostają zapisane do plików dla gracza pierwszego oraz drugiego.

| Gracz1 | Gracz2 |
|--------------|--------------|
| ----- | ----- |
| Punkty: 20 | Punkty: 10 |
| Statki: | Statki: |
| [#][#][L][#] | [X][X][#][#] |
| [#][X][X][#] | [#][#][#][#] |
| [#][#][#][#] | [#][#][#][#] |
| [#][#][#][#] | [X][#][#][#] |
| Trafenia: | Trafenia: |
| [X][X][#][#] | [#][#][#][#] |
| [#][#][#][#] | [#][X][X][#] |
| [#][#][#][#] | [#][#][#][#] |
| [X][#][#][#] | [#][#][#][#] |

6. Wnioski

Napisanie tego programu oraz nauczenie się rozumowania obiektowego w zaledwie ponad miesiąc było lekkim wyzwaniem. Program działa i spełnia wszystkie podane elementy. Uważam, że gdyby było więcej czasu na napisanie projektu to udało by mi się go bardziej rozbudować np. dodać bibliotekę graficzną. Z uwagi, że jeszcze nigdy nie miałem styczności z bibliotekami graficznymi postanowiłem zrealizować projekt w konsoli Visuala. Największym problemem było sensownie dodanie operatorów do programu i ich wykorzystanie, ponieważ na początku nie widziałem sensu ich użycia. Ostatecznie to zagadnienie zostało zrealizowane i wykorzystane w kodzie.

Statki

Generated by Doxygen 1.8.16

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

| | |
|-----------------------------|----|
| Gracz | ?? |
| Mapa | ?? |
| SprawdzPoprawnosc | ?? |
| Lodka | ?? |
| Statek | ?? |
| Prom | ?? |
| Transportowiec | ?? |
| wygladMap | ?? |

Chapter 2

Class Index

2.1 Class List

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

| | | |
|-----------------------------------|-------|----|
| Gracz | | ?? |
| Lodka | | ?? |
| Mapa | | ?? |
| Prom | | ?? |
| SprawdzPoprawnosc | | ?? |
| Statek | | ?? |
| Transportowiec | | ?? |
| wygladMap | | ?? |

Chapter 3

File Index

3.1 File List

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

| | |
|--|----|
| C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/ Funkcje.h | ?? |
| C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/ Gracz.h | ?? |
| C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/ Mapa.h | ?? |
| C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/ RodzajeStatkow.h | ?? |
| C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/ StrukturyDanych.h | ?? |

Chapter 4

Class Documentation

4.1 Gracz Class Reference

```
#include <Gracz.h>
```

Public Member Functions

- [Gracz](#) ([Mapa](#) mapa, int pkt)
- [~Gracz](#) ()
- int [zwrocPunkty](#) ()
- void [edytujPunkty](#) (int pkt)
- void [ZapiszDoPlikuGracza1](#) (fstream &plik)
- void [ZapiszDoPlikuGracza2](#) (fstream &plik)
- [Gracz](#) & [operator+=](#) (int pkt)
- void [zarzadzajRozgrywka](#) ([Gracz](#) grac2, fstream &g1, fstream &g2)

Public Attributes

- int [punktacja](#)
punkty gracza
- [Mapa rozstawienia](#)
mapa statkow gracza

Friends

- ostream & [operator<<](#) (ostream &stream, const [Gracz](#) &s)

4.1.1 Detailed Description

Klasa reprezentujaca gracza

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Gracz()

```
Gracz::Gracz (
    Mapa mapa,
    int pkt )
```

Konstruktor dwuargumentowy klasy [Gracz](#)

Parameters

| | |
|-------------|----------------------------------|
| <i>mapa</i> | obiekt typu Mapa |
| <i>pkt</i> | punktacja gracza |

4.1.2.2 ~Gracz()

```
Gracz::~~Gracz ( )
```

Desturktor

4.1.3 Member Function Documentation

4.1.3.1 edytujPunkty()

```
void Gracz::edytujPunkty (
    int pkt )
```

Metoda edytuje punktacje gracza

Parameters

| | |
|------------|---------------|
| <i>pkt</i> | punkty gracza |
|------------|---------------|

4.1.3.2 operator+=()

```
Gracz & Gracz::operator+= (
    int pkt )
```

Operator += do edycji punktacji

Parameters

| | |
|------------|-------------------|
| <i>pkt</i> | punkty do dodania |
|------------|-------------------|

Returns

gracz

4.1.3.3 ZapiszDoPlikuGracza1()

```
void Gracz::ZapiszDoPlikuGracza1 (  
    fstream & plik )
```

Metoda zapisu dla gracza pierwszego

Parameters

| | |
|-------------|----------------|
| <i>plik</i> | plik do zapisu |
|-------------|----------------|

4.1.3.4 ZapiszDoPlikuGracza2()

```
void Gracz::ZapiszDoPlikuGracza2 (  
    fstream & plik )
```

Metoda zapisu dla gracza drugiego

Parameters

| | |
|-------------|----------------|
| <i>plik</i> | plik do zapisu |
|-------------|----------------|

4.1.3.5 zarzadzajRozgrywka()

```
void Gracz::zarzadzajRozgrywka (  
    Gracz gracz2,  
    fstream & g1,  
    fstream & g2 )
```

Metoda odpowiedzialna za zarzadzanie rozgrywka

Parameters

| | |
|---------------|-----------|
| <i>gracz2</i> | gracz |
| <i>g1</i> | plik nr 1 |
| <i>g2</i> | plik nr 2 |

4.1.3.6 zwrocPunkty()

```
int Gracz::zwrocPunkty ( )
```

Metoda zwraca punkty gracza

Returns

punkty

4.1.4 Friends And Related Function Documentation

4.1.4.1 operator<<

```
ostream& operator<< (
    ostream & stream,
    const Gracz & s ) [friend]
```

Operator << do wypisywania punktacji

Parameters

| | |
|---------------|-----------------------------------|
| <i>stream</i> | stream |
| <i>s</i> | obiekt typu Gracz |

Returns

stream

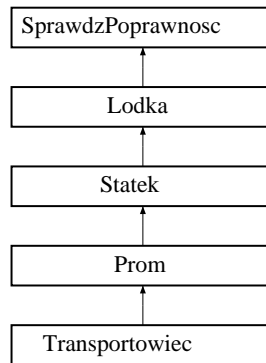
The documentation for this class was generated from the following files:

- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/[Gracz.h](#)
- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/Gracz.cpp

4.2 Lodka Class Reference

```
#include <RodzajeStatkow.h>
```

Inheritance diagram for Lodka:



Public Member Functions

- [Lodka](#) (string sym, int roz, int i1, char c1)
- [~Lodka](#) ()
- virtual bool [poprawnosc](#) (int roz, int i1, char c1)
- virtual void [zapisz](#) (int roz, int i1, char s1, bool [poprawnosc](#))
- int [zwrocX](#) ()
- char [zwrocY](#) ()
- string [zwrocSymbol](#) ()
- [Lodka zworcLodke](#) (char roz)

Protected Attributes

- int [icoo1](#)
piewsza wspolrzeczna x
- char [ccoo1](#)
pierwsza wspolrzeczna y

Private Attributes

- string [symbol](#)
symbol lodki
- int [rozmiar](#)
rozmiar mapy

4.2.1 Detailed Description

Klasa reprezentująca obiekty typu [Lodka](#), dziedziczy z klasy [SprawdzPoprawnosc](#)

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Lodka()

```
Lodka::Lodka (
    string sym,
    int roz,
    int i1,
    char c1 )
```

Konstruktor wieloargumentowy klasy [Lodka](#)

Parameters

| | |
|-------------|----------------------|
| <i>sym</i> | symbol lodki |
| <i>roz</i> | rozmiar mapy |
| <i>int</i> | wspolrzedna x statku |
| <i>char</i> | wspolrzedna y statku |

4.2.2.2 ~Lodka()

```
Lodka::~~Lodka ( )
```

Destruktor

4.2.3 Member Function Documentation

4.2.3.1 poprawnosc()

```
bool Lodka::poprawnosc (
    int roz,
    int i1,
    char c1 ) [virtual]
```

Wirtualna metoda, ktora sprawdza poprawnosc danych

Parameters

| | |
|-------------|----------------------|
| <i>roz</i> | rozmiar mapy |
| <i>int</i> | wspolrzedna y statku |
| <i>char</i> | wspolrzedna x statku |

Returns

bool informacja o poprawnosci danych wejsciowych

Implements [SprawdzPoprawnosc](#).

4.2.3.2 zapisz()

```
void Lodka::zapisz (
    int roz,
    int il,
    char s1,
    bool poprawnosc ) [virtual]
```

Wirtualna metoda, ktora zapisuje dane do obiektu typu [Lodka](#)

Parameters

| | |
|-------------------|---|
| <i>roz</i> | rozmiar mapy |
| <i>int</i> | wspolrzeczna y statku |
| <i>char</i> | wspolrzeczna x statku |
| <i>poprawnosc</i> | informacja o poprawnosci danych wejsciowych |

Implements [SprawdzPoprawnosc](#).

Reimplemented in [Transportowiec](#), and [Prom](#).

4.2.3.3 zworcLodke()

```
Lodka Lodka::zworcLodke (
    char roz )
```

Metoda zwraca obiekt typu [Lodka](#)

Parameters

| | |
|------------|--------------|
| <i>roz</i> | rozmiar mapy |
|------------|--------------|

Returns

lodka

4.2.3.4 zwrocSymbol()

```
string Lodka::zwrocSymbol ( )
```

Metoda zwraca wspolrzeczna Y

Returns

symbol

4.2.3.5 zwrocX()

```
int Lodka::zwrocX ( )
```

Metoda zwraca wspolrzeczna X

Returns

wspolrzeczna X

4.2.3.6 zwrocY()

```
char Lodka::zwrocY ( )
```

Metoda zwraca wspolrzeczna Y

Returns

wspolrzeczna Y

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

- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/[RodzajeStatkow.h](#)
- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/RodzajeStatkow.cpp

4.3 Mapa Class Reference

```
#include <Mapa.h>
```

Public Member Functions

- [Mapa](#) (string s, int r, [wygladMap](#) *&mS, [wygladMap](#) *&mT, [Lodka](#) l, [Statek](#) sta, [Prom](#) p, [Transportowiec](#) t)
- [~Mapa](#) ()
- string [zwrocSymbolMapy](#) ()
- int [zwrocrozmiar](#) ()
- [wygladMap](#) *& [zwrocMapeStatkow](#) ()
- [wygladMap](#) *& [zwrocMapeTrafien](#) ()
- void [zapiszMape](#) ([wygladMap](#) *&stat)
- void [wypiszMape](#) ([wygladMap](#) *statLUBtraf)
- void [zapiszMapeStrzalow](#) ([wygladMap](#) *&traf)
- int [strzal](#) ([wygladMap](#) *&statkiPrzeciwnika, [wygladMap](#) *&mapaStrzalowGracza)
- void [przejdzDalej](#) ()
- void [wypiszDoPliku](#) (fstream &plik, [wygladMap](#) *mS, [wygladMap](#) *mT)
- void [zwalnianiePamieci](#) ([wygladMap](#) *&pHead)

Protected Attributes

- string [symbol](#)
symbol wyswietlania pola mapy
- int [rozmiar](#)
rozmiar mapy (4,6,10)
- [wygladMap](#) * [mapaStatkow](#)
mapa statkow gracza
- [wygladMap](#) * [mapaTrafien](#)
mapa trafien gracza
- [Lodka](#) [lodka](#)
lodka
- [Statek](#) [statek](#)
statek
- [Prom](#) [prom](#)
prom
- [Transportowiec](#) [trans](#)
transportowiec

4.3.1 Detailed Description

Klasa mapy gry

4.3.2 Constructor & Destructor Documentation

4.3.2.1 Mapa()

```
Mapa::Mapa (
    string s,
    int r,
    wygladMap *& mS,
    wygladMap *& mT,
    Lodka l,
    Statek sta,
    Prom p,
    Transportowiec t ) [inline]
```

Lista inicjalizacyjna mapy

4.3.2.2 ~Mapa()

```
Mapa::~~Mapa ( )
```

destruktor

4.3.3 Member Function Documentation

4.3.3.1 przejdzDalej()

```
void Mapa::przejdzDalej ( )
```

Metoda odpowiada za przejście do następnego ruchu

4.3.3.2 strzal()

```
int Mapa::strzal (
    wygladMap *& statkiPrzeciwnika,
    wygladMap *& mapaStrzalowGracza )
```

Metoda odpowiedzialna za strzał gracza

Parameters

| | |
|---------------------------|--------------------|
| <i>statkiPrzeciwnika</i> | statki przeciwnika |
| <i>mapaStrzalowGracza</i> | statki gracza |

Returns

punkty za trafienie

4.3.3.3 wypiszDoPliku()

```
void Mapa::wypiszDoPliku (
    fstream & plik,
    wygladMap * mS,
    wygladMap * mT )
```

Metoda zapisuje statki oraz trafienia gracza do pliku tekstowego

Parameters

| | |
|-------------|------------------|
| <i>plik</i> | plik do zapisu |
| <i>mS</i> | statki gracza |
| <i>mT</i> | trafienia gracza |

4.3.3.4 wypiszMape()

```
void Mapa::wypiszMape (
    wygladMap * statLUBtraf )
```

Metoda wypisuje statki lub trafienia gracza

Parameters

| | |
|--------------------|-------------------------------|
| <i>statLUBtraf</i> | mapa statkow lub mapa trafien |
|--------------------|-------------------------------|

4.3.3.5 zapiszMape()

```
void Mapa::zapiszMape (
    wygladMap *& stat )
```

Metoda zapisuje statki gracza

Parameters

| | |
|-------------|--------------|
| <i>stat</i> | mapa statkow |
|-------------|--------------|

4.3.3.6 zapiszMapeStrzalow()

```
void Mapa::zapiszMapeStrzalow (
    wygladMap *& traf )
```

Metoda zapisuje trafienia gracza

Parameters

| | |
|-------------|------------------|
| <i>traf</i> | trafienia gracza |
|-------------|------------------|

4.3.3.7 zwalnianiePamieci()

```
void Mapa::zwalnianiePamieci (
    wygladMap *& pHead )
```

Metoda odpowiada za usuniecie list statkow oraz trafien

Parameters

| | |
|--------------|------------------------------------|
| <i>pHead</i> | poczatek listy statkow lub trafien |
|--------------|------------------------------------|

4.3.3.8 zworcSymbolMapy()

```
string Mapa::zworcSymbolMapy ( )
```

Metoda zwraca symbol mapy

Returns

symbol mapy

4.3.3.9 zwrocMapeStatkow()

```
wygladMap *& Mapa::zwrocMapeStatkow ( )
```

Metoda zwraca mape statkow

Returns

mapa statkow

4.3.3.10 zwrocMapeTrafien()

```
wygladMap *& Mapa::zwrocMapeTrafien ( )
```

Metoda zwraca mape trafien

Returns

mapa trafien

4.3.3.11 zwrocrozmiar()

```
int Mapa::zwrocrozmiar ( )
```

Metoda zwraca rozmiar mapy

Returns

rozmiar mapy

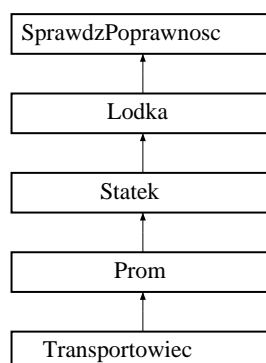
The documentation for this class was generated from the following files:

- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/[Mapa.h](#)
- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/Mapa.cpp

4.4 Prom Class Reference

```
#include <RodzajeStatkow.h>
```

Inheritance diagram for Prom:



Public Member Functions

- [Prom](#) (string sym, int roz, int i1, char c1, int i2, char c2, int i3, char c3, [Lodka](#) l, [Statek](#) s)
- [~Prom](#) ()
- virtual bool [poprawnosc](#) (int roz, int i3, char c3, [Statek](#) s)
- virtual void [zapisz](#) (int roz, int i3, char c3, bool [poprawnosc](#))
- int [zwrocX3](#) ()
- char [zwrocY3](#) ()
- [Prom](#) [zwrocProm](#) (char roz, [Lodka](#) lodka, [Statek](#) statek)

Protected Attributes

- int [icoo3](#)
trzecia wspolrzeczna x
- char [ccoo3](#)
trzecia wspolrzeczna y

4.4.1 Detailed Description

Klasa reprezentujaca obiekty typu [Prom](#), dziedziczy z klasy [Statek](#)

4.4.2 Constructor & Destructor Documentation

4.4.2.1 Prom()

```
Prom::Prom (
    string sym,
    int roz,
    int i1,
    char c1,
    int i2,
    char c2,
    int i3,
    char c3,
    Lodka l,
    Statek s )
```

Konstruktor wieloargumentowy klasy [Statek](#), podczas wywołania tego konstruktora, wywołują się konstruktory klasy [Statek](#) oraz klasy [Lodka](#)

Parameters

| | |
|------------|------------------------------------|
| <i>sym</i> | symbol promu |
| <i>roz</i> | rozmiar mapy |
| <i>i1</i> | pierwsza wspolrzeczna x statku |
| <i>c1</i> | pierwsza wspolrzeczna y statku |
| <i>i2</i> | druga wspolrzeczna x statku |
| <i>c2</i> | druga wspolrzeczna y statku |
| <i>i3</i> | trzecia wspolrzeczna x statku |
| <i>c3</i> | trzecia wspolrzeczna y statku |
| <i>l</i> | obiekt typu Lodka |
| <i>s</i> | obiekt typu Statek |

4.4.2.2 ~Prom()

```
Prom::~~Prom ( )
```

Destruktor

4.4.3 Member Function Documentation

4.4.3.1 poprawnosc()

```
bool Prom::poprawnosc (
    int roz,
    int i3,
    char c3,
    Statek s ) [virtual]
```

Wirtualna metoda, ktora sprawdza poprawnosc danych

Parameters

| | |
|------------|-----------------------------|
| <i>roz</i> | rozmiar mapy |
| <i>i3</i> | druga wspolrzeczna y statku |
| <i>c3</i> | druga wspolrzeczna x statku |
| <i>s</i> | obiekt typu statek |

Returns

bool informacja o poprawnosci danych wejscowych

4.4.3.2 zapisz()

```
void Prom::zapisz (
    int roz,
    int i3,
    char c3,
    bool poprawnosc ) [virtual]
```

Wirtualna metoda, ktora zapisuje dane do obiektu typu [Prom](#)

Parameters

| | |
|-------------------|--|
| <i>roz</i> | rozmiar mapy |
| <i>i3</i> | druga wspolrzeczna y statku |
| <i>c3</i> | druga wspolrzeczna x statku |
| <i>poprawnosc</i> | informacja o poprawnosci danych wejscowych |

Reimplemented from [Lodka](#).

Reimplemented in [Transportowiec](#).

4.4.3.3 zwrocProm()

```
Prom Prom::zwrocProm (
    char roz,
    Lodka lodka,
    Statek statek )
```

Metoda zwraca obiekt typu [Prom](#)

Parameters

| | |
|---------------|------------------------------------|
| <i>roz</i> | rozmiar mapy |
| <i>lodka</i> | obiekt typu Lodka |
| <i>statek</i> | obiekt typu Statek |

Returns

[Prom](#)

4.4.3.4 zwrocX3()

```
int Prom::zwrocX3 ( )
```

Metoda zwraca trzecia wspolrzeczna X

Returns

druga wspolrzeczna X

4.4.3.5 zwrocY3()

```
char Prom::zwrocY3 ( )
```

Metoda zwraca trzecia wspolrzeczna Y

Returns

wspolrzeczna druga Y

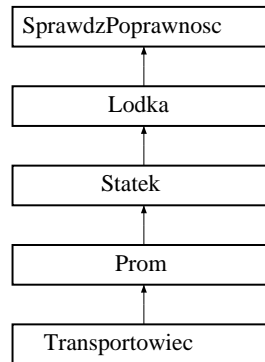
The documentation for this class was generated from the following files:

- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/[RodzajeStatkow.h](#)
- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/[RodzajeStatkow.cpp](#)

4.5 SprawdzPoprawnosc Class Reference

```
#include <RodzajeStatkow.h>
```

Inheritance diagram for SprawdzPoprawnosc:



Public Member Functions

- virtual bool [poprawnosc](#) (int roz, int i1, char c1)=0
- virtual void [zapisz](#) (int roz, int i1, char s1, bool [poprawnosc](#))=0

4.5.1 Detailed Description

Wirtualna klasa do sprawdzania poprawnosci i zapisu danych do obiektow typu [Lodka](#)

4.5.2 Member Function Documentation

4.5.2.1 [poprawnosc\(\)](#)

```
virtual bool SprawdzPoprawnosc::poprawnosc (  
    int roz,  
    int i1,  
    char c1 ) [pure virtual]
```

Wirtualna metoda, ktora sprawdza poprawnosc danych

Parameters

| | |
|-------------|----------------------|
| <i>roz</i> | rozmiar mapy |
| <i>int</i> | wspolrzedna y statku |
| <i>char</i> | wspolrzedna x statku |

Returns

bool informacja o poprawnosci danych wejsciowych

Implemented in [Lodka](#).

4.5.2.2 zapisz()

```
virtual void SprawdzPoprawnosc::zapisz (  
    int roz,  
    int i1,  
    char s1,  
    bool poprawnosc ) [pure virtual]
```

Wirtualna metoda, ktora zapisuje dane do obiektu typu [Lodka](#)

Parameters

| | |
|-------------------|---|
| <i>roz</i> | rozmiar mapy |
| <i>int</i> | wspolrzeczna y statku |
| <i>char</i> | wspolrzeczna x statku |
| <i>poprawnosc</i> | informacja o poprawnosci danych wejsciowych |

Implemented in [Transportowiec](#), [Prom](#), and [Lodka](#).

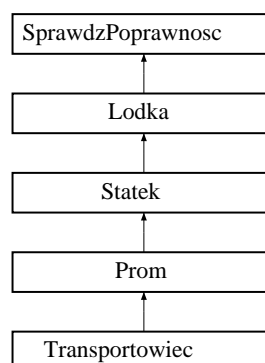
The documentation for this class was generated from the following file:

- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/[RodzajeStatkow.h](#)

4.6 Statek Class Reference

```
#include <RodzajeStatkow.h>
```

Inheritance diagram for Statek:



Public Member Functions

- [Statek](#) (string *sym*, int *roz*, int *i1*, char *c1*, int *i2*, char *c2*, [Lodka](#) *l*)
- [~Statek](#) ()
- virtual bool [poprawnosc](#) (int *roz*, int *i2*, char *c2*, [Lodka](#) *l*)
- virtual void [zapisz](#) (int *i2*, char *c2*, bool [poprawnosc](#))
- int [zwrocX2](#) ()
- char [zwrocY2](#) ()
- [Statek](#) [zworcStatek](#) (char *roz*, [Lodka](#) *lodka*)

Protected Attributes

- int [icoo2](#)
druga wspolrzeczna x
- char [ccoo2](#)
druga wspolrzeczna y

4.6.1 Detailed Description

Klasa reprezentująca obiekty typu [Statek](#), dziedziczy z klasy [Lodka](#)

4.6.2 Constructor & Destructor Documentation

4.6.2.1 Statek()

```
Statek::Statek (
    string sym,
    int roz,
    int i1,
    char c1,
    int i2,
    char c2,
    Lodka l )
```

Konstruktor wieloargumentowy klasy [Statek](#), podczas wywołania tego konstruktora, wywołuje się konstruktor klasy [Lodka](#)

Parameters

| | |
|------------|-----------------------------------|
| <i>sym</i> | symbol statku |
| <i>roz</i> | rozmiar mapy |
| <i>i1</i> | pierwsza wspolrzeczna x statku |
| <i>c1</i> | pierwsza wspolrzeczna y statku |
| <i>i2</i> | druga wspolrzeczna x statku |
| <i>c2</i> | druga wspolrzeczna y statku |
| <i>l</i> | obiekt typu Lodka |

4.6.2.2 ~Statek()

```
Statek::~~Statek ( )
```

Destruktor

4.6.3 Member Function Documentation

4.6.3.1 poprawnosc()

```
bool Statek::poprawnosc (
    int roz,
    int i2,
    char c2,
    Lodka l ) [virtual]
```

Wirtualna metoda, ktora sprawdza poprawnosc danych

Parameters

| | |
|------------|-----------------------------------|
| <i>roz</i> | rozmiar mapy |
| <i>i2</i> | druga wspolrzeczna y statku |
| <i>c2</i> | druga wspolrzeczna x statku |
| <i>l</i> | obiekt typu Lodka |

Returns

bool informacja o poprawnosci danych wejsciowych

4.6.3.2 zapisz()

```
void Statek::zapisz (
    int i2,
    char c2,
    bool poprawnosc ) [virtual]
```

Wirtualna metoda, ktora zapisuje dane do obiektu typu [Statek](#)

Parameters

| | |
|-------------------|---|
| <i>roz</i> | rozmiar mapy |
| <i>i2</i> | druga wspolrzeczna y statku |
| <i>c2</i> | druga wspolrzeczna x statku |
| <i>poprawnosc</i> | informacja o poprawnosci danych wejsciowych |

4.6.3.3 zwrocStatek()

```
Statek Statek::zwrocStatek (
    char roz,
    Lodka lodka )
```

Metoda zwraca obiekt typu [Statek](#)

Parameters

| | |
|--------------|-----------------------------------|
| <i>roz</i> | rozmiar mapy |
| <i>lodka</i> | obiekt typu Lodka |

Returns

statek

4.6.3.4 zwrocX2()

```
int Statek::zwrocX2 ( )
```

Metoda zwraca druga wspolrzedna X

Returns

druga wspolrzedna X

4.6.3.5 zwrocY2()

```
char Statek::zwrocY2 ( )
```

Metoda zwraca druga wspolrzedna Y

Returns

wspolrzedna druga Y

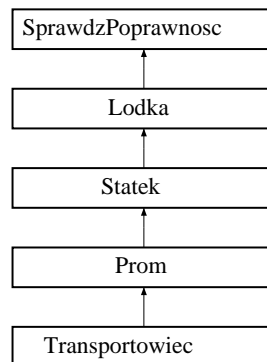
The documentation for this class was generated from the following files:

- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/[RodzajeStatkow.h](#)
- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/[RodzajeStatkow.cpp](#)

4.7 Transportowiec Class Reference

```
#include <RodzajeStatkow.h>
```

Inheritance diagram for Transportowiec:



Public Member Functions

- [Transportowiec](#) (string sym, int roz, int i1, char c1, int i2, char c2, int i3, char c3, int i4, char c4, [Lodka](#) l, [Statek](#) s, [Prom](#) p)
- [~Transportowiec](#) ()
- virtual bool [poprawnosc](#) (int roz, int i4, char c4, [Prom](#) s)
- virtual void [zapisz](#) (int roz, int i4, char c4, bool [poprawnosc](#))
- int [zwrocX4](#) ()
- char [zwrocY4](#) ()
- [Transportowiec zworcTransportowiec](#) (char roz, [Lodka](#) lodka, [Statek](#) statek, [Prom](#) prom)

Protected Attributes

- int [icoo4](#)
czwarta wspolrzeczna x
- char [ccoo4](#)
czwarta wspolrzeczna y

4.7.1 Detailed Description

Klasa reprezentująca obiekty typu [Transportowiec](#), dziedziczy z klasy [Prom](#)

4.7.2 Constructor & Destructor Documentation

4.7.2.1 Transportowiec()

```
Transportowiec::Transportowiec (
    string sym,
    int roz,
    int i1,
    char c1,
    int i2,
    char c2,
    int i3,
    char c3,
    int i4,
    char c4,
    Lodka l,
    Statek s,
    Prom p )
```

Konstruktor wieloargumentowy klasy [Statek](#), podczas wywołania tego konstruktora, wywołują się konstruktory klasy [Statek](#) oraz klasy [Lodka](#)

Parameters

| | |
|------------|------------------------------------|
| <i>sym</i> | symbol transportowca |
| <i>roz</i> | rozmiar mapy |
| <i>i1</i> | pierwsza współrzędna x statku |
| <i>c1</i> | pierwsza współrzędna y statku |
| <i>i2</i> | druga współrzędna x statku |
| <i>c2</i> | druga współrzędna y statku |
| <i>i3</i> | trzecia współrzędna x statku |
| <i>c3</i> | trzecia współrzędna y statku |
| <i>i4</i> | czwarta współrzędna x statku |
| <i>c4</i> | czwarta współrzędna y statku |
| <i>l</i> | obiekt typu Lodka |
| <i>s</i> | obiekt typu Statek |
| <i>p</i> | obiekt typu Prom |

4.7.2.2 ~Transportowiec()

```
Transportowiec::~~Transportowiec ( )
```

Destruktor

4.7.3 Member Function Documentation

4.7.3.1 poprawnosc()

```
bool Transportowiec::poprawnosc (
    int roz,
    int i4,
    char c4,
    Prom s ) [virtual]
```

Wirtualna metoda, ktora sprawdza poprawnosc danych

Parameters

| | |
|------------|----------------------------------|
| <i>roz</i> | rozmiar mapy |
| <i>i3</i> | druga wspolrzeczna y statku |
| <i>c3</i> | druga wspolrzeczna x statku |
| <i>s</i> | obiekt typu Prom |

Returns

bool informacja o poprawnosci danych wejscowych

4.7.3.2 zapisz()

```
void Transportowiec::zapisz (
    int roz,
    int i4,
    char c4,
    bool poprawnosc ) [virtual]
```

Wirtualna metoda, ktora zapisuje dane do obiektu typu [Transportowiec](#)

Parameters

| | |
|-------------------|--|
| <i>roz</i> | rozmiar mapy |
| <i>i4</i> | czwarta wspolrzeczna y statku |
| <i>c4</i> | czwarta wspolrzeczna x statku |
| <i>poprawnosc</i> | informacja o poprawnosci danych wejscowych |

Reimplemented from [Prom](#).

4.7.3.3 zworcTransportowiec()

```
Transportowiec Transportowiec::zworcTransportowiec (
    char roz,
    Lodka lodka,
```

```
Statek statek,  
Prom prom )
```

Metoda zwraca obiekt typu [Transportowiec](#)

Parameters

| | |
|---------------|------------------------------------|
| <i>roz</i> | rozmiar mapy |
| <i>lodka</i> | obiekt typu Lodka |
| <i>statek</i> | obiekt typu Statek |
| <i>prom</i> | obiekt typu Prom |

Returns

transportowiec

4.7.3.4 zwrocX4()

```
int Transportowiec::zwrocX4 ( )
```

Metoda zwraca czwarta wspolrzeczna X

Returns

czwarta wspolrzeczna X

4.7.3.5 zwrocY4()

```
char Transportowiec::zwrocY4 ( )
```

Metoda zwraca czwarta wspolrzeczna Y

Returns

wspolrzeczna czwarta Y

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

- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/[RodzajeStatkow.h](#)
- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/RodzajeStatkow.cpp

4.8 wygladMap Struct Reference

```
#include <StrukturyDanych.h>
```

Public Attributes

- string [symbol](#)
symbol wyświetlania pola mapy
- int [rozmiar](#)
rozmiar mapy
- bool [statek](#)
czy statek jest na danym miejscu czy go nie ma
- string [nazwaStatku](#)
nazwa statku
- int [y](#)
- char [x](#)
<wspolrzedna y
- [wygladMap](#) * [next](#)
adres nastepnego elementu listy

4.8.1 Detailed Description

< napis przechowywany w liscie (nazwa pliku) adres nastepnego elementu listy struktura listy jednokierunkowej

4.8.2 Member Data Documentation

4.8.2.1 x

char `wygladMap::x`

<wspolrzedna y

wspolrzedna x

The documentation for this struct was generated from the following file:

- C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/[StrukturyDanych.h](#)

Chapter 5

File Documentation

5.1 C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/Funkcje.h File Reference

```
#include <iostream>
#include "Gracz.h"
#include "Mapa.h"
#include "RodzajeStatkow.h"
#include "StrukturyDanych.h"
```

Functions

- void [instrukcja](#) ()
- char [podajRozmiar](#) ()
- int [zwrocPrawdziwyRozmiar](#) (char fakerozmiar)

5.1.1 Function Documentation

5.1.1.1 instrukcja()

```
void instrukcja ( )
```

Funkcja wypisuje instrukcje dla uzytkownikow.

5.1.1.2 podajRozmiar()

```
char podajRozmiar ( )
```

Funkcja odpowiada za wprowadzenie uproszczonego rozmiaru podanego przez użytkownika

Returns

uproszczony rozmiar

5.1.1.3 zwrocPrawdziwyRozmiar()

```
int zwrocPrawdziwyRozmiar (
    char fakerozmiar )
```

Funkcja zamienia uproszczony rozmiar mapy rozgrywki, na prawdziwy.

Parameters

| | |
|--------------------|--------------------------|
| <i>fakerozmiar</i> | uproszczony rozmiar mapy |
|--------------------|--------------------------|

Returns

prawdziwy rozmiar mapy

5.2 C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/↵ Statki/Gracz.h File Reference

```
#include <iostream>
#include "Mapa.h"
#include <fstream>
```

Classes

- class [Gracz](#)

5.3 C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/↵ Statki/Mapa.h File Reference

```
#include <iostream>
#include "StrukturyDanych.h"
#include "RodzajeStatkow.h"
#include <fstream>
```


Classes

- class [Mapa](#)

5.4 C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/RodzajeStatkow.h File Reference

```
#include <iostream>
```

Classes

- class [SprawdzPoprawnosc](#)
- class [Lodka](#)
- class [Statek](#)
- class [Prom](#)
- class [Transportowiec](#)

5.5 C:/Users/Adam/Desktop/PK/195abe22-gr22-repo/Projekt/Statki/Statki/StrukturyDanych.h File Reference

```
#include <iostream>
```

Classes

- struct [wygladMap](#)