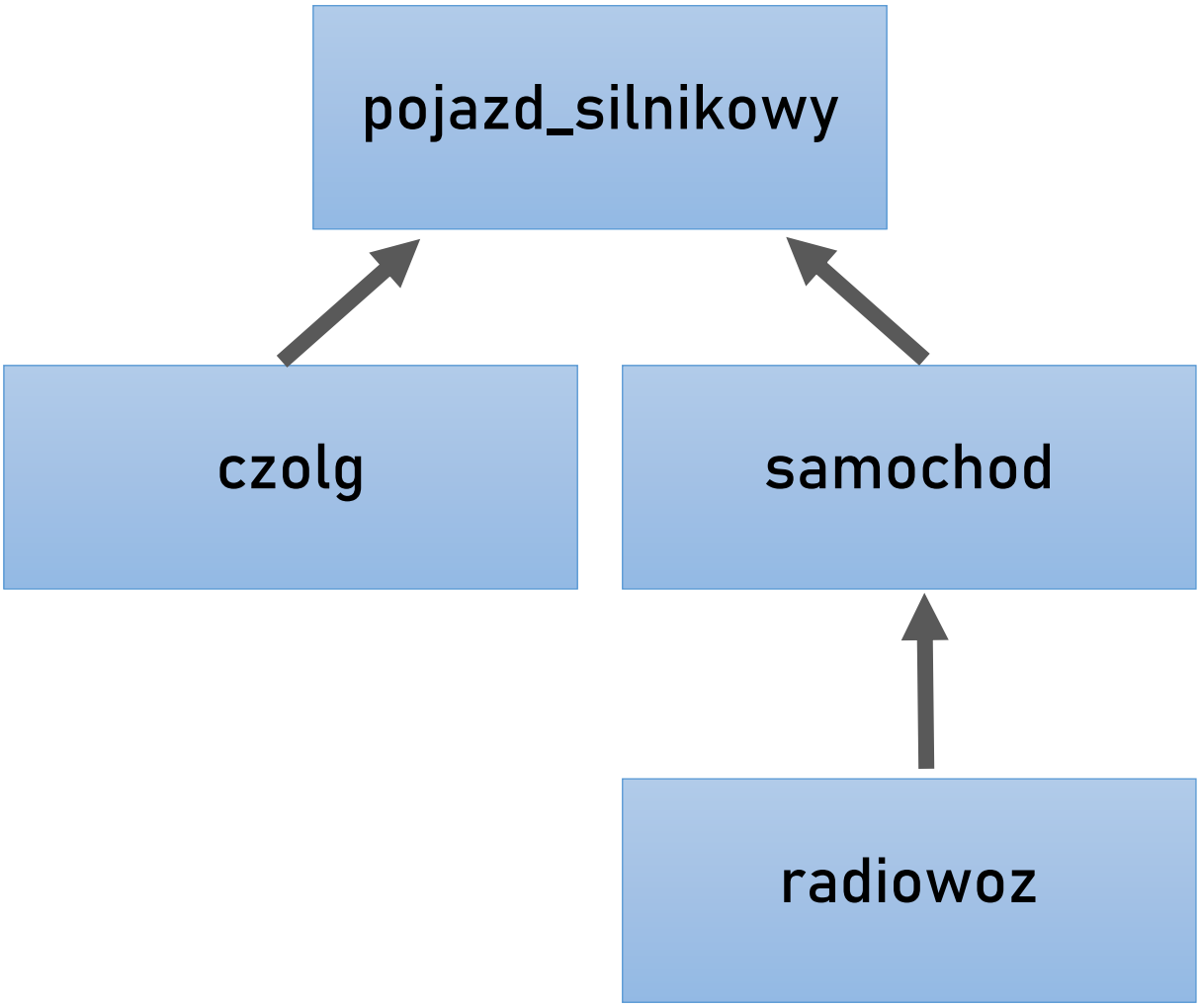
Bogusław Malewski
PROI Projekt 2

Rozdział 1 – Wprowadzenie

Schemat dziedziczenia:



Pojazd_silnikowy jest klasą abstrakcyjną i stanowi klasę bazową dla wszystkich pozostałych klas głównych. Zawiera metody czysto wirtualne:
virtual int in_info() = 0; //wpisuje podstawowe dane
virtual int out_info() = 0; //wypisuje podstawowe dane
virtual int fin_info(ifstream &plik) = 0; //wczytuje stan obiektu do pliku
virtual int fout_info(ofstream &plik) = 0; //zapisuje stan obiektu do pliku

Klasa czołg dziedziczy po abstrakcyjnej klasie bazowej pojazd_silnikowy.

Klasa samochod z poprzedniego projektu została zmieniona. Zachowała swoje wszystkie metody i składowe, ale teraz dziedziczy po abstrakcyjnej klasie bazowej pojazd_silnikowy.

Klasa radiowoz dziedziczy po klasie samochod.

Rozdział 2 – Opis interfejsu klas

```
class pojazd_silnikowy
{
private:
    string marka;
    string model;
    int rocznik = 0;
    double spalanie_na_100km = 9;
    kierowca * kierowca_1 = nullptr;
    bak bak_1;
    silnik silnik_1;
public:
    pojazd_silnikowy(); //konstruktor
    virtual ~pojazd_silnikowy(); //destruktor
    pojazd_silnikowy(const pojazd_silnikowy& orig);
//konstruktor kopiujący
    virtual pojazd_silnikowy &operator=(const pojazd_silnikowy
& sam1); //operator przypisania

    virtual int in_info() = 0; //wpisuje podstawowe dane
    virtual int out_info() = 0; //wypisuje podstawowe dane
    virtual int fin_info(ifstream &plik) = 0; //wczytuje stan
obiektu do pliku
    virtual int fout_info(ofstream &plik) = 0; //zapisuje stan
obiektu do pliku

    int in_kierowca(); //wprowadza kierowcę do pojazdu
    int out_kierowca(); //wypisuje informacje o kierowcy
    int usun_kierowca(); //usuwa kierowcę z pojazdu
    bool out_prawo_jazdy(); //zwraca true jeśli kierowca ma
prawo jazdy

    int in_bak(); //wpisuje ilość paliwa
    int out_bak(); //zwraca i wypisuje ilość paliwa w baku

    int on_silnik(); //włącza silnik
    int off_silnik(); //wyłącza silnik

    int przejedz(double kilometry); //przejeżdża daną liczbę
kilometrów

    bool operator++(); //włącza silnik
    bool operator--(); //wyłącza silnik
    friend ostream & operator<<(ostream & os, pojazd_silnikowy
&pojazd_silnikowy1);
    friend istream & operator>>(istream & is, pojazd_silnikowy
&pojazd_silnikowy1);
    friend ofstream & operator<<(ofstream & os,
pojazd_silnikowy &pojazd_silnikowy1);
    friend ifstream & operator>>(ifstream & is,
pojazd_silnikowy &pojazd_silnikowy1);
```

```
};

class czolg : public pojazd_silnikowy
{
private:
    enum {min_zaloga = 2};
    int amunicja = 20;
    bool zaladowany = false;
    pasazer zaloga[4];
public:
    czolg();
    czolg(const czolg& orig);
    virtual ~czolg();

    int in_info(); //Wpisuje podstawowe dane
    int out_info(); //Wypisuje podstawowe dane
    int fin_info(ifstream &plik);//wczytuje stan obiektu z
pliku
    int fout_info(ofstream &plik); //zapisuje stan obiektu do
pliku
    int dodaj_amunicje(int ilosc);//Dodaje wpisaną ilość
amunicji
    int laduj();//Ładuje armatę czołgu
    int strzelaj();//Strzela z armaty używając załadowanej
amunicji

    int dodaj_zaloge(int miejsce, string imie, string
nazwisko, int wiek);//Dodaje członka załogi
    int zwolnij_zaloge(int miejsce);//Zwalnia członka załogi
    int out_zaloga(int miejsce);//Wypisuje dane członka załogi
z danego miejsca
    int out_all_zaloga();//Wypisuje dane wszystkich członków
załogi

    friend ostream & operator<<(ostream & os, czolg &czolg1);
    friend istream & operator>>(istream & is, czolg &czolg1);
    friend ofstream & operator<<(ofstream & os, czolg
&czolg1);
    friend ifstream & operator>>(ifstream & is, czolg
&czolg1);
};
```

```
class samochod : public pojazd_silnikowy
{
private:
    int miejsca_dla_pasazerow;
    pasazer * pasazerowie = nullptr;
    bagaznik bagaznik_1;
public:
    samochod(int miejsca = 0); //konstruktor
    ~samochod(); //destruktor
    samochod(const samochod & sam1); //konstruktor kopiujący
    samochod &operator=(const samochod & sam1); //operator
przypisania

    int in_info(); //wpisuje podstawowe dane
    int out_info(); //wypisuje podstawowe dane
    int fin_info(ifstream &plik); //wczytuje stan obiektu z
pliku
    int fout_info(ofstream& plik); //zapisuje stan obiektu do
pliku

    int in_pasazer(int miejsce); //wprowadza pasażera na dane
miejsce
    int out_pasazer(int miejsce); //wypisuje dane pasażera z
danego miejsca
    int out_all_pasazer(); //wypisuje dane wszystkich
pasażerów z auta
    int usun_pasazer(int miejsce); //usuwa pasażera z danego
miejsca
    int usun_all_pasazer(); //usuwa wszystkich pasażerów z
auta

    int open_bagaznik(); //otwiera bagażnik
    int close_bagaznik(); //zamyka bagażnik
    int zaladuj_bagaznik(); //ładuje bagażnik
    int rozladuj_bagaznik(); //rozładowuje bagżnik

    int przejedz(double kilometry); //przejeżdża daną liczbę
kilometrów

    friend ostream & operator<<(ostream & os, samochod
&samochod1);
    friend istream & operator>>(istream & is, samochod
&samochod1);
    friend ofstream & operator<<(ofstream & os, samochod
&samochod1);
    friend ifstream & operator>>(ifstream & os, samochod
&samochod1);
};
```

```
class radiowoz : public samochod
{
private:
    bool syg_swietlna;
    bool syg_dzwiekowa;
    pasazer zatrzymani[3];

public:
    radiowoz();
    radiowoz(const radiowoz& orig);
    virtual ~radiowoz();

    int in_info(); //wpisuje podstawowe dane
    int out_info(); //wypisuje podstawowe dane
    int fin_info(ifstream &plik);//wczytuje stan obiektu z
pliku
    int fout_info(ofstream& plik); //zapisuje stan obiektu do
pliku

    int on_dzwiek(); //Włącza syganlizację świetlną
    int off_dzwiek(); //Wyłącza syganlizację świetlną

    int on_swiatlo(); //Włącza syganlizację dźwiękową
    int off_swiatlo(); //Wyłącza syganlizację dźwiękową

    int zatrzymaj(int miejsce, string imie, string nazwisko,
int wiek);//Wprowadza do radiowozu zatrzymaną osobę
    int zwolnij(int miejsce); //Wyprowadza z radiowozu
zatrzymaną osobę
    int out_zatrzymani(int miejsce); //Wypisuje dane
zatrzymanej osoby
    int out_all_zatrzymani(); //Wypisuje dane wszystkich
zatrzymanych osob

    friend ostream & operator<<(ostream & os, radiowoz
&radiowoz1);
    friend istream & operator>>(istream & is, radiowoz
&radiowoz1);
    friend ofstream & operator<<(ofstream & os, radiowoz
&radiowoz1);
    friend ifstream & operator>>(ifstream & is, radiowoz
&radiowoz1);
};
```

Rozdział 3 – Opis funkcji main

Funkcja main umożliwia zarządzanie wszystkimi 3 klasami za pomocą konsolowego interfejsu. Interfejs jest niewrażliwy na błędy użytkownika i umożliwia użycie wszystkich dostępnych metod wszystkich klas.
Funkcja main umożliwia zapis i wczytanie stanu obiektu wszystkich klas do/z pliku
czolg.txt
samochod.txt
radiowoz.txt

Na koniec program zawsze wypisuje końcowy stan wszystkich obiektów do konsoli za pomocą tablicy wskaźników na obiekty klasy bazowej klasy pojazd_silnikowy, korzystając z metod wirtualnych.