МИНОБРНАУКИ РФ

ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»

Кафедра «Программное обеспечение»

Отчет

по лабораторной работе № 1

по объектно-ориентированному программированию

Вариант №1

Выполнил студент группы Б04-191-3: Р.А. Гумметов

Принял: К.С. Чернышев

Ижевск 2019

Постановка задачи

Определить класс Person:

закрытые поля:

имя - строка ( 20 символов )

фамилия - строка ( 20 символов )

отчество - строка ( 20 символов )

возраст - целое число

открытые поля:

Конструктор без параметров, устанавливающий значения свойство по умолчанию.

Конструктор с 4 параметрами строка, строка, строка, число, для инициализации полей (имя, фамилия, отчество, возраст) класса.

Метод для установки возраста персоны, с проверкой на правильность ввода ( возраст >= 0 и <= 200 )

Метод для установки имени персоны, проверять длину строки.

Метод для установки фамилии персоны, проверять длину строки.

Метод для установки отчества персоны, проверять длину строки.

Метод для получения возраста персоны.

Метод для получения имени персоны.

Метод для получения фамилии персоны.

Метод для получения отчества персоны.

Метод вывода информации о персоне на экран.

Определить перечисляемый тип(enum) марка\_автомобиля

Определить класс Автомобиль

закрытые поля:

водитель - поле типа Person

марка - поле типа марка\_автомобиля

номер автомобиля - строка ( 9 символов )

открытые поля:

Конструктор устанавливающий свойства объекта по умолчанию.

Конструктор с 2мя параметрами марка и номер автомобиля.

Метод для установки водителя автомобиля, с проверкой возраста водителя, если возраст < 18 выводить предупреждение.

Метод для вывода информации об автомобиле: марка, номер, фио водителя.

В методе Main()

Создать 2 экземпляра класса Person.

Вывести о них информацию на экран.

Создать экземпляр класса автомобиль.

Задать классу автомобиль водителя.

Вывести информацию об автомобиле.

Текст программы

#include <iostream>

#include <string>

using namespace std;

class Person

{

private:

string FirstName;

string LastName;

string Patronymic;

int Age;

public:

Person()

{

FirstName = "Ivan";

LastName = "Ivanov";

Patronymic = "Ivanovich";

Age = 25;

}

Person(string firstName, string lastName, string patronymic, int age)

{

SetFirstName(firstName);

SetLastName(lastName);

SetPatronymic(patronymic);

SetAge(age);

}

void SetFirstName(string firstName)

{

if (firstName.length() <= 20 && firstName.length() > 0)

FirstName = firstName;

}

void SetLastName(string lastName)

{

if (lastName.length() <= 20 && lastName.length() > 0)

LastName = lastName;

}

void SetPatronymic(string patronymic)

{

if (patronymic.length() <= 20 && patronymic.length() > 0)

Patronymic = patronymic;

}

void SetAge(int age)

{

if (age >= 0 && age <= 200)

Age = age;

}

string GetFirstName()

{

return FirstName;

}

string GetLAstName()

{

return LastName;

}

string GetPatronymic()

{

return Patronymic;

}

int GetAge()

{

return Age;

}

void OutPutOnScreen()

{

cout << "Имя: " << FirstName << endl;

cout << "Фамилия: " << LastName << endl;

cout << "Отчество: " << Patronymic << endl;

cout << "Возраст: " << Age << endl;

}

};

enum CarBrand

{

VOLKSWAGEN,

AUDI,

RENAULT,

BENTLEY,

TOYOTA

};

class Car

{

private:

Person Driver;

CarBrand Brand;

string Number;

public:

Car()

{

Driver = Person();

Brand = VOLKSWAGEN;

Number = "A001AA";

}

Car(CarBrand brand, string number)

{

if (brand >= 0 && brand <= 4)

Brand = brand;

else

Brand = VOLKSWAGEN;

if (number.length() > 0 && number.length() <= 9)

Number = number;

else

Number = "A001AA";

}

void SetDriver(Person driver)

{

Driver.SetFirstName(driver.GetFirstName());

Driver.SetLastName(driver.GetLAstName());

Driver.SetPatronymic(driver.GetPatronymic());

if (driver.GetAge() < 18)

cout << "Водителю " << driver.GetLAstName() << " " << driver.GetFirstName() << " " << driver.GetPatronymic() << " нет 18 лет" << endl;

else

Driver.SetAge(driver.GetAge());

}

void DriverOnScreen()

{

cout << "Марка автомобиля: ";

switch (Brand)

{

case VOLKSWAGEN:

cout << "VOLKSWAGEN" << endl;

break;

case AUDI:

cout << "AUDI" << endl;

break;

case RENAULT:

cout << "RENAULT" << endl;

break;

case BENTLEY:

cout << "BENTLEY" << endl;

break;

case TOYOTA:

cout << "TOYOTA" << endl;

break;

default:

break;

}

cout << "Номер автомобиля: " << Number << endl;

cout << "Фамилия: " << Driver.GetLAstName() << endl;

cout << "Имя: " << Driver.GetFirstName() << endl;

cout << "Отчество: " << Driver.GetPatronymic() << endl;

}

};

int main()

{

setlocale(LC\_ALL, "rus");

Person man1;

man1.OutPutOnScreen();

cout << endl << endl;

Person man2{ "Владислав" , "Цепнев" , "Евгеньевич" , 30 };

man2.OutPutOnScreen();

cout << endl << endl;

Car car1{ BENTLEY , "m777mm" };

car1.SetDriver(man2);

car1.DriverOnScreen();

}

Пример работы программы

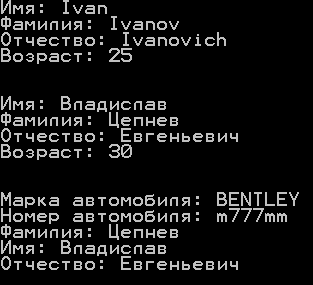


Рис. 1

Вопросы:

1) Какая основная особенность объектно-ориентированного программирования?

ООП позволяет использовать один и тот же программный код с разными данными. На основе классов создается множество объектов, у каждого из которых могут быть собственные значения полей. Нет необходимости вводить множество переменных.

2) Что такое класс?

Класс – это способ описания сущности, определяющий состояние и поведение, зависящее от этого состояния, а также правила для взаимодействия с данной сущностью

3) Для каких целей используется конструктор?

Конструктор используется для инициализации начального состояния класса при создании его экземпляра.

4) Что такое перечисляемый тип данных?

Перечисляемый тип данных - это такой тип данных, множество значений которого представляет собой ограниченный список идентификаторов.

5)  Что такое инкапсуляция, и как она применяется в лабораторной работе?

Инкапсуляция - это механизм, который объединяет данные и код, манипулирующий этими данными, а также защищает и то, и другое от внешнего вмешательства или неправильного использования. В лабораторной работе, например, переменные FirstName, LastName, Patronymic, Age и т.д. являются закрытыми и обратиться к ним можно только внутри класса. Чтобы работать с ними, в классе определены методы GetFirstName(), SetFirstName() и т.д.