**ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №13**

**Тема: Наследование»**

***Цель работы:*** Изучить возможности наследования классов на языке С++.

**Задание:** Выполнить задание согласно варианту, используя множественное наследование:

1. Используя родительский класс «ТРАНСПОРТ» породить производный класс «АВТОМОБИЛЬ». Используя классы «ВОДИТЕЛЬ» и «АВТОМОБИЛЬ», описать класс «ВОДИТЕЛЬ АВТОМОБИЛЯ». Расширить класс «ВОДИТЕЛЬ АВТОМОБИЛЯ» создав два производных класса «ВОДИТЕЛЬ СЛУЖЕБНОГО АВТОМОБИЛЯ» и «ВОДИТЕЛЬ ТАКСИ». Продумать для данной иерархии классов поля и методы (обязательно: вывод информации о водителе, автомобиле)

2. Используя родительский класс «СЛУЖАЩИЙ» породить производный класс «ДИРЕКТОР». Используя классы «ФИРМА» и «ДИРЕКТОР», описать класс «РУКОВОДИТЕЛЬ ФИРМЫ». Расширить класс «РУКОВОДИТЕЛЬ ФИРМЫ» создав два производных класса «РУКОВОДИТЕЛЬ ГОС.УЧ.» и «РУКОВОДИТЕЛЬ ООО». Продумать для данной иерархии классов поля и методы (обязательно: вывод информации о фирме и руководителе)

3. Используя родительский класс «НЕДВИЖИМОСТЬ» породить производный класс «ЗДАНИЕ». Используя классы «ЗДАНИЕ» и «ВЛАДЕЛЕЦ», описать класс «ВЛАДЕЛЕЦ ЗДАНИЯ». Расширить класс «ВЛАДЕЛЕЦ ЗДАНИЯ» создав два производных класса «ВЛАДЕЛЕЦ ЧАСТНОГО ДОМА» и «ВЛАДЕЛЕЦ ОТЕЛЯ». Продумать для данной иерархии классов поля и методы (обязательно: вывод информации о владельце и здании)

4. Используя родительский класс «ТРАНСПОРТ» породить производный класс «САМОЛЕТ». Используя классы «ПИЛОТ» и «САМОЛЕТ», описать класс «ПИЛОТ САМОЛЕТА». Расширить класс «ПИЛОТ САМОЛЕТА» создав два производных класса «ПИЛОТ ГРАЖДАНСКОГО САМОЛЕТА» и «ПИЛОТ ВОЕННОГО САМОЛЕТА». Продумать для данной иерархии классов поля и методы (обязательно: вывод информации о пилоте, самолете)

5. Используя родительский класс «РАБОТНИК» породить производный класс «РЕЖИССЕР». Используя классы «ФИЛЬМ» и «РЕЖИССЕР», описать класс «РЕЖИССЕР ФИЛЬМА». Расширить класс «РЕЖИССЕР ФИЛЬМА» создав два производных класса «РЕЖИССЕР ХУДОЖЕСТВЕННОГО ФИЛЬМА» и «РЕЖИССЕР ДОКУМЕНТАЛЬНОГО ФИЛЬМА». Продумать для данной иерархии классов поля и методы (обязательно: вывод информации о фирме и руководителе).

***Ход выполнения***

Т.к. мой вариант – 1, я создал 6 классов, которые наследуют друг друга.

Класс Transport (листинг 1) имеет следующие поля:

1. int num\_of\_wheels – кол-во колёс,
2. int num\_of\_doors – кол-во дверей,
3. bool has\_engine – определяет наличие двигателя,
4. int wheel\_type – тип руля (0-мотоциклетный, 1-автомобильный, 3-рычаги). По сути содержит индекс элемента массива wheels[3][27],
5. int num\_of\_seats – кол-во мест для сидения,
6. int weight – масса транспортного средства,
7. int max\_weight – максимально допустимая масса ТС,
8. dimensions dimens – объект структуры dimensions, содержащий габариты ТС.

Все эти поля объявлены в protected, чтобы они были доступны потомственным классам.

Также данный класс имеет следующие методы:

1. Transport() и Transport(int whel, int doors, bool eng, int whtype, int seats, int weig, int mweigh, float h, float w, float d) – являются конструкторами класса, которые не принимают и принимают параметры соответственно,
2. void set(int whel, int doors, bool eng, int whtype, int seats, int weig, int mweigh, float h, float w, float d) – изменяет данные класса,
3. void print\_transport() – выводит данные объекта в cout. Наличие двигателя выводится как: h\_eng[has\_engine], т.е. если 0 – отсутствует, 1 – есть. Тип руля выводится как: wheels[wheel\_type].

Листинг 1 – Класс Transport

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21.  22.  23.  24.  25.  26.  27.  28.  29.  30.  31.  32.  33.  34.  35.  36.  37.  38.  39.  40.  41.  42.  43.  44.  45.  46. | class Transport{  public:  Transport(){}  Transport(int whel, int doors, bool eng, int whtype, int seats, int weig, int mweigh, float h, float w, float d){  num\_of\_wheels = whel;  num\_of\_doors = doors;  has\_engine = eng;  wheel\_type = whtype;  num\_of\_seats = seats;  weight = weig;  max\_weight = mweigh;  dimens = {h, w, d};  }  void set(int whel, int doors, bool eng, int whtype, int seats, int weig, int mweigh, float h, float w, float d){  num\_of\_wheels = whel;  num\_of\_doors = doors;  has\_engine = eng;  wheel\_type = whtype;  num\_of\_seats = seats;  weight = weig;  max\_weight = mweigh;  dimens = {h, w, d};  }  void print\_transport(){  cout << "Транстпорт: \n"  << " Кол-во колёс: " << num\_of\_wheels << endl  << " Кол-во дверей: " << num\_of\_doors << endl  << " Наличие двигателя: "<< h\_eng[has\_engine]<< endl  << " Тип руля: " << wheels[wheel\_type] << endl  << " Мест для сидения: " << num\_of\_seats << endl  << " Масса: " << weight << endl  << " Максимальная масса: " << max\_weight << endl  << " Габариты:\n высота:" << dimens.h << endl  << " ширина: " << dimens.w << endl  << " глубина: " << dimens.d << endl;  }  protected:  int num\_of\_wheels = 0;  int num\_of\_doors = 0;  bool has\_engine = 0;  int wheel\_type = 0; // 0-мотоциклетный, 1-автомобильный, 3-рычаги  int num\_of\_seats = 1;  int weight = 1;  int max\_weight = 10;  dimensions dimens = {1, 1, 1};  }; |

Класс Car (листинг 2) наследуется от Transport и следовательно имеет все его публичные и протектные методы и свойства.

Помимо доступных полей класса Transport здесь также в области protected объявлены следующие свойства:

1. char \*number – строка с гос. номером,
2. int vehicle\_passport – номер ПТС,
3. int cert\_of\_reg – Номер СТС.

Методы:

1. void \_init(int whel, int doors, bool eng, int whtype, int seats, int weig, int mweigh, float h, float w, float d, char \*num, int pasp, int cert) – приватный метод инициализации переменных, задаёт данные как для свойств данного класса, так и для свойств родительского класса,
2. Car() – конструктор, без параметров, передаёт в \_init стандартные данные любого автомобиля,
3. Car(int whel, int doors, bool eng, int whtype, int seats, int weig, int mweigh, float h, float w, float d, char \*num, int pasp, int cert) – конструктор, принимающий данные,
4. void set\_car(int whel, int doors, bool eng, int whtype, int seats, int weig, int mweigh, float h, float w, float d, char \*num, int pasp, int cert) – публичный метод, изменяющий данные объекта,
5. void print\_car(bool print\_trans) – публичный метод, принимающий логическое значение и выводящий на экран данные из текущего и родительского класса, если переданный параметр равен правде, то данные родительского класса выводятся, иначе нет.

Листинг 2 – Класс Car

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21.  22.  23.  24.  25.  26.  27.  28.  29.  30.  31.  32.  33.  34.  35.  36.  37.  38.  39. | class Car: protected Transport{  public:  Car(){  \_init(4, 2, 1, 1, 2, 400, 500, 1, 2.5, 3, "А111АА111", 0, 0);  }  Car(int whel, int doors, bool eng, int whtype, int seats, int weig, int mweigh, float h, float w, float d, char \*num, int pasp, int cert){  \_init(whel, doors, eng, whtype, seats, weig, mweigh, h, w, d, num, pasp, cert);  }  void set\_car(int whel, int doors, bool eng, int whtype, int seats, int weig, int mweigh, float h, float w, float d, char \*num, int pasp, int cert){  \_init(whel, doors, eng, whtype, seats, weig, mweigh, h, w, d, num, pasp, cert);  }  void print\_car(bool print\_trans){  if(print\_trans){  print\_transport();  }  cout << "Автомобиль: \n"  << " Гос. номер: " << number << endl  << " ПТС: " << vehicle\_passport << endl  << " СТС: " << cert\_of\_reg << endl;  }  private:  void \_init(int whel, int doors, bool eng, int whtype, int seats, int weig, int mweigh, float h, float w, float d, char \*num, int pasp, int cert){  num\_of\_wheels = whel;  num\_of\_doors = doors;  has\_engine = eng;  wheel\_type = whtype;  num\_of\_seats = seats;  weight = weig;  max\_weight = mweigh;  dimens = {h, w, d};  number = num;  vehicle\_passport = pasp;  cert\_of\_reg = cert;  }  protected:  char \*number = "А111АА111";  int vehicle\_passport = 0;  int cert\_of\_reg = 0;  }; |

Класс Driver (листинг 3) имеет следующие свойства:

1. char \*name – имя водителя,
2. int age – возраст водителя,
3. categ categories – объект структуры categ, который хранит список открытых категорий,
4. int year\_obtaining\_dr\_lic – год, когда были получены первые права,
5. int year\_last\_dr\_lic\_change – год, когда последний раз они были поменяны.

А также методы:

1. Driver(char\* nm, int ag, int year\_of\_obt, int year\_of\_last) – конструктор,
2. void set\_driver(char\* nm, int ag, int year\_of\_obt, int year\_of\_last) – сеттер, изменяющий имя, возраст, а также года,
3. void set\_driver(char\* nm, int ag) – сеттер, изменяющий только имя и возраст,
4. void set\_ctg(char \*ctg, bool val) – меняет категорию, принимает строку с названием категории и логическое значение,
5. void print\_driver() – выводит данные класса,
6. void \_show\_ctgs() – приватный метод, выводит открытые категории.

Листинг 3 – Класс Driver

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21.  22.  23.  24.  25.  26.  27.  28.  29.  30.  31.  32.  33.  34.  35.  36.  37.  38.  39.  40.  41.  42.  43.  44.  45.  46.  47.  48.  49.  50.  51.  52.  53.  54.  55.  56.  57.  58.  60.  61.  62.  63.  64.  65.  66.  67.  68.  69.  70.  71.  72.  73.  74.  75.  76.  77.  78.  79.  80. | class Driver{  public:  char \*name = "";  int age = 16;  categ categories = {0, 0, 0, 0, 0, 0, 0};  Driver(char\* nm, int ag, int year\_of\_obt, int year\_of\_last){  name = nm;  age = ag;  year\_obtaining\_dr\_lic = year\_of\_obt;  year\_last\_dr\_lic\_change = year\_of\_last;  }  Driver(){}  ~Driver(){}  void set\_driver(char\* nm, int ag, int year\_of\_obt, int year\_of\_last){  name = nm;  age = ag;  year\_obtaining\_dr\_lic = year\_of\_obt;  year\_last\_dr\_lic\_change = year\_of\_last;  }  void set\_driver(char\* nm, int ag){  name = nm;  age = ag;  }  void set\_ctg(char \*ctg, bool val){  if (ctg == "a"){  categories.a = val;  }else if(ctg == "b"){  categories.b = val;  }else if(ctg == "c"){  categories.c = val;  }else if(ctg == "d"){  categories.d = val;  }else if(ctg == "be"){  categories.be = val;  }else if(ctg == "ce"){  categories.ce = val;  }else if(ctg == "de"){  categories.de = val;  }else{  cout << "Неверный параметр: Такой категории нет.\n";  }  }  void print\_driver(){  time\_t now = time(0);  tm \*ltm = localtime(&now);  int now\_year = 1900 + ltm->tm\_year;  cout << "Водитель: \n Имя: " << name << endl << " Возраст: " << age << endl <<  " Стаж: " << now\_year - year\_obtaining\_dr\_lic << endl << " Открытые категории: ";  \_show\_ctgs();  }  private:  void \_show\_ctgs(){  if(categories.a){  cout << " A ";  }  if(categories.b){  cout << " B ";  }  if(categories.c){  cout << " C ";  }  if(categories.d){  cout << " D ";  }  if(categories.be){  cout << " BE ";  }  if(categories.ce){  cout << " CE ";  }  if(categories.de){  cout << " DE ";  }  cout << endl;  }  protected:  int year\_obtaining\_dr\_lic = 1900;  int year\_last\_dr\_lic\_change = 2000;  }; |

Класс Car\_Driver (листинг 4) наследуется от классов Driver и Car.

Методы:

1. Car\_Driver(char\* nm, int ag, int year\_of\_obt, int year\_of\_last) – конструктор,
2. void print\_car\_driver(bool print\_cr, bool print\_dr) – выводит данные объекта, принимает два логических параметра: первый отвечает за вывод данных автомобиля, второй за вывод данных водителя.

Листинг 4 – Класс Car\_Driver

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18. | class Car\_Driver: public Driver, public Car{  public:  Car\_Driver(char\* nm, int ag, int year\_of\_obt, int year\_of\_last){  name = nm;  age = ag;  year\_obtaining\_dr\_lic = year\_of\_obt;  year\_last\_dr\_lic\_change = year\_of\_last;  }  Car\_Driver(){}  void print\_car\_driver(bool print\_cr, bool print\_dr){  if(print\_cr){  print\_car(1);  }  if(print\_dr){  print\_driver();  }  }  protected:  }; |

Класс Official\_Car\_Driver (листинг 5) наследуется от класса Car\_Driver и имеет только одно свойство: char \*company – строка с названием компании.

А также методы:

1. Official\_Car\_Driver(char\* nm, int ag, int year\_of\_obt, int year\_of\_last, char \*comp) – конструктор,
2. void set\_company(char \*comp) – изменяет компанию,
3. void print\_off\_car\_driver(bool print\_cr) – выводит данные объекта,
4. void set\_category(char \*ctg, bool val) – изменяет категорию.

Листинг 5 – Класс Official\_Car\_Driver

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21. | class Official\_Car\_Driver: protected Car\_Driver{  public:  char \*company = "";  Official\_Car\_Driver(char\* nm, int ag, int year\_of\_obt, int year\_of\_last, char \*comp){  name = nm;  age = ag;  year\_obtaining\_dr\_lic = year\_of\_obt;  year\_last\_dr\_lic\_change = year\_of\_last;  company = comp;  }  void set\_company(char \*comp){  company = comp;  }  void print\_off\_car\_driver(bool print\_cr){  print\_car\_driver(print\_cr, 1);  cout << " Компания: " << company << endl;  }  void set\_category(char \*ctg, bool val){  set\_ctg(ctg, val);  }  }; |

Класс Taxi\_Car\_Driver (листинг 6) наследуется от класса Car\_Driver и имеет свойства:

1. float rating – рейтинг водителя,
2. int taxi\_experience – опыт работы в такси,
3. char \*taxi\_company – таксопарк.

А также методы:

1. Taxi\_Car\_Driver(char\* nm, int ag, int year\_of\_obt, int year\_of\_last, float rat, int exp, char \*comp) – конструктор,
2. void print\_taxi\_car\_driver(bool print\_cr) – выводит данные объекта, принимает логический параметр, который определяет будут ли выводиться данные машины,
3. void set\_category(char \*ctg, bool val) – изменяет категорию.

Листинг 6 – Класс Taxi\_Car\_Driver

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21.  22.  23.  24. | class Taxi\_Car\_Driver: protected Car\_Driver{  public:  float rating = 0;  int taxi\_experience = 0;  char \*taxi\_company = "";  Taxi\_Car\_Driver(char\* nm, int ag, int year\_of\_obt, int year\_of\_last, float rat, int exp, char \*comp){  name = nm;  age = ag;  year\_obtaining\_dr\_lic = year\_of\_obt;  year\_last\_dr\_lic\_change = year\_of\_last;  rating = rat;  taxi\_experience = exp;  taxi\_company = comp;  }  void print\_taxi\_car\_driver(bool print\_cr){  print\_car\_driver(print\_cr, 1);  cout << " Рейтинг: " << rating << endl  << " Стаж таксиста: " << taxi\_experience << endl  << " Таксопарк: " << taxi\_company << endl;  }  void set\_category(char \*ctg, bool val){  set\_ctg(ctg, val);  }  }; |

1. Функция main (листинг 7) реализует использование объектов классов. Вывод представлен на рисунке 1.

Листинг 7 – Main

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21.  22.  23.  24.  25. | int main(){  setlocale(LC\_ALL, "ru\_RU.utf8");  // Transport tran;  // tran.print\_transport();  Car car;  car.set\_car(4, 5, 1, 1, 5, 920, 1400, 1.2, 3, 4.6, "А320МС134", 1389462364, 856942357);  // car.print\_car(1);  categ ctgs={0,1,0,0,1,0,0};  char \*nm = "Иван";  Driver dr(nm, 25, 2016, 2012);  // Driver dr;  dr.set\_ctg("b", 1);  dr.set\_ctg("be", 1);  // dr.print\_driver();  Car\_Driver cdr(car, nm, 25, 2018, 2013);  cdr.set\_ctg("a", 1);  // cdr.print\_car\_driver(1, 1);  Official\_Car\_Driver ocdr(car, "Влад", 30, 2015, 2020, "S2 INK");  ocdr.set\_category("b", 1);  ocdr.print\_off\_car\_driver(0);  Taxi\_Car\_Driver tcdr(car, "Владимир", 30, 2015, 2020, 4.3, 2, "S2 INK");  tcdr.set\_category("b", 1);  tcdr.print\_taxi\_car\_driver(0);  return 0;  } |

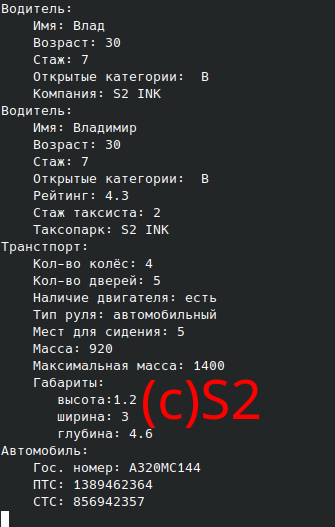


Рисунок 1 – Вывод в консоле