Лабораторна робота 5(1)

1)

#include <iostream>

using namespace std;

class Triad {

public:

virtual bool isLeapYear(int y) = 0;

public:

int day;

int month;

int year;

};

class Date: public Triad {

public:

Date() {

day = 01;

month = 01;

year = 2000;

}

Date(int d, int m, int y) {

day = d;

month = m;

year = y;

}

void setDay(int d) {

day = d;

}

void setMonth(int m) {

month = m;

}

void setYear(int y) {

year = y;

}

int getDay() {

return day;

}

int getMonth() {

return month;

}

int getYear() {

return year;

}

int days\_in\_month[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

bool isLeapYear(int year) {

if (((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0)) {

return true;

}

return false;

}

bool checkDay(Date &date) {

for (auto i: { 0, 2, 4, 6, 7, 9, 11 }) {

if (date.getMonth() == i + 1) {

if (date.getDay() >= 1 && date.getDay() <= days\_in\_month[i]) {

return true;

}

}

}

for (auto i: { 3, 5, 8, 10 }) {

if (date.getMonth() == i + 1) {

if (date.getDay() >= 1 && date.getDay() <= days\_in\_month[i]) {

return true;

}

}

}

if (date.getMonth() == 2) {

if (isLeapYear(date.getYear())) {

if (date.getDay() >= 1 && date.getDay() <= 29) {

return true;

}

} else {

if (date.getDay() >= 1 && date.getDay() <= 28) {

return true;

}

}

}

return false;

}

bool checkMonth(Date &date) {

if (date.getMonth() >= 1 && date.getMonth() <= 12) {

return true;

}

return false;

}

bool checkYear(Date &date) {

if (date.getYear() >= 1900 && date.getYear() <= 2040) {

return true;

}

return false;

}

void checkDate(Date &date) {

if ((checkDay(date) == true) &&

(checkMonth(date) == true) &&

(checkYear(date) == true)) {

cout << "This date is correct!\n";

} else {

cout << "This date is NOT correct!\n";

}

}

};

int main(void) {

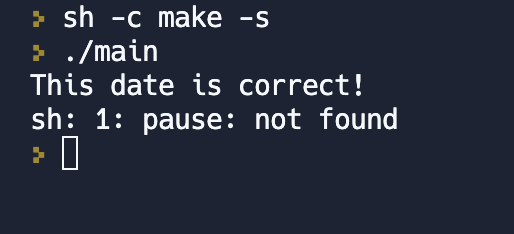
Date obj(31, 05, 2000);

obj.checkDate(obj);

system("pause");

return 0;

}



2)

#include <iostream>

#include <string>

struct dvyhun

{

int power{};

};

struct automobile

{

std::string marka;

double cina{};

dvyhun pow;

};

struct vant: public automobile

{

int vantag{};

vant()=default;

vant(int vantag,const automobile& a):automobile(a),vantag(vantag){}

void OutputVant()const

{

std::cout<<"Марка: "<<marka<<'\n';

std::cout<<"Ціна: "<<cina<<'\n';

std::cout<<"Сила: "<<pow.power<<'\n';

std::cout<<"Вантажопідйомність: "<<vantag<<'\n';

}

};

int main()

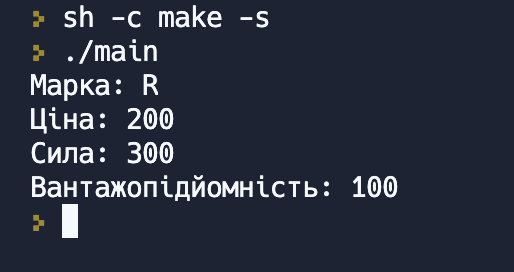
{

vant obj(100, automobile{"R", 200, {300}});

obj.OutputVant();

return 0;

}



3)

#include<iostream>

#include<vector>

#include<string>

class Wheel

{

private:

unsigned rad;

public:

Wheel()

: rad(0)

{}

Wheel(unsigned radius)

: rad(radius)

{}

void set\_rad(unsigned radius)

{

this->rad = radius;

}

unsigned get\_rad() const

{

return this->rad;

}

};

class Car

{

private:

std::vector<Wheel> car\_wheel;

std::string firma\_name;

public:

Car()

: firma\_name("unknow\_firma")

{}

Car(unsigned kol\_wheel, unsigned rad\_wheel, std::string firma\_name)

: firma\_name(firma\_name)

{

for (int i = 0; i < kol\_wheel; ++i)

car\_wheel.push\_back(Wheel(rad\_wheel));

}

void set\_firma(std::string name)

{

this->firma\_name = name;

}

std::string get\_firma\_name() const

{

return this->firma\_name;

}

std::vector<Wheel> get\_wheel() const

{

return this->car\_wheel;

}

};

class HeavyCar : public Car

{

private:

unsigned capacity;

public:

HeavyCar(unsigned kol\_wheel, unsigned rad\_wheel, std::string firma\_name, unsigned car\_capacity)

: Car(kol\_wheel, rad\_wheel, firma\_name)

, capacity(car\_capacity)

{}

void set\_capacity(unsigned car\_capacity)

{

this->capacity = car\_capacity;

}

unsigned get\_capacity() const

{

return this->capacity;

}

};

int main()

{

Wheel wheel\_1(5);

std::cout << "Радіус колеса = " << wheel\_1.get\_rad() << std::endl;

wheel\_1.set\_rad(10);

std::cout << "Діаметр = " << wheel\_1.get\_rad() << std::endl;

std::cout << std::endl;

Car t\_2109(4,13,"Tesla");

std::cout << "Фірма " << t\_2109.get\_firma\_name() << std::endl;

std::cout << "Радіус колеса = " << t\_2109.get\_wheel()[0].get\_rad() << std::endl;

std::cout << std::endl;

HeavyCar GMC(10, 20, "GMC", 20);

std::cout << "Фірма " << GMC.get\_firma\_name() << std::endl;

std::cout << "Радіус колеса = " << GMC.get\_wheel()[0].get\_rad() << std::endl;

std::cout << "Ємність = " << GMC.get\_capacity() << std::endl;

std::system("pause");

return 0;

}

