

Lab 6.3(1.1)

Code:

#include <stdexcept>

#include <limits>

#include <iostream>

using namespace std;

char slash = '/';

char backslash = 92;

class theOrigin // стартовий класс виводу фігури

{

public:

virtual void print()

{

cout << " " << slash << backslash << "\n" << slash << slash << backslash << backslash << endl;

}

};

class art1 : public theOrigin

{

void print() override

{

cout << " " << slash << backslash << "\n" << slash << "\*\*" << backslash << endl;

}

};

class art2 : public theOrigin

{

void print() override

{

cout << " " << slash << backslash << "\n" << slash << "++" << backslash << endl;

}

};

class poly // Поліформізм всіх класів

{

public:

void print(theOrigin\* A)

{

A->print();

}

};

int main(){

theOrigin a;

art1 b;

art2 c;

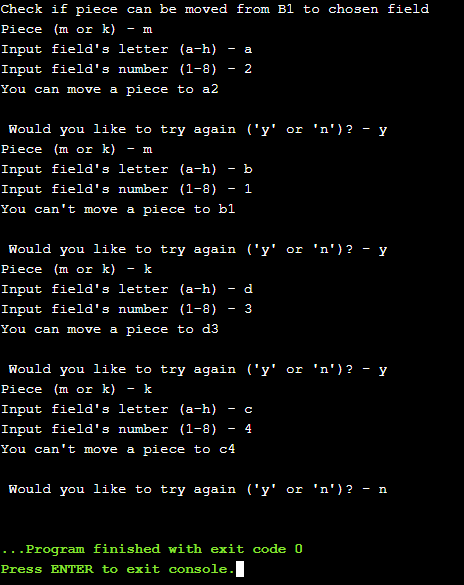
theOrigin\* ptr[3] = { &a, &b, &c }; // масив указників всіх об єктів дочірніх класів

poly polymorph; // об єк классу поліформності

for (unsigned int i = 0; i < 3; i++) //цикл, для всіх елементів масиву polymorph.print(ptr[i]); // виклик методу print для кожного з цих об єктів

return 0;

}



Lab 6.3(1.2)

#include <stdexcept>

#include <limits>

#include <iostream>

using namespace std;

char slash = '/';

char backslash = 92;

class Piece

{

public:

virtual void Check(char letter, int number) = 0;

};

class Men :public Piece

{

public:

void Check(char letter, int number) override

{

if ((letter == 'a' && number == 2) || (letter == 'c' && number == 2))

cout << "You can move a piece to " << letter << number << endl;

else

cout << "You can't move a piece to " << letter << number << endl;

}

};

class Kings :public Piece

{

void Check(char letter, int number) override

{

if ((letter == 'a' && number == 2) || (letter == 'c' && number == 2) || (letter == 'd' && number == 3) || (letter == 'e' && number == 4) || (letter == 'f' && number == 5) || (letter == 'g' && number == 6) || (letter == 'h' && number == 7))

cout << "You can move a piece to " << letter << number << endl;

else

cout << "You can't move a piece to " << letter << number << endl;

}

};

int main(){

char piece, letter, command;

int number;

cout << "Check if piece can be moved from B1 to chosen field" << endl;

do // перевірити точки

{

cout << "Piece (m or k) - "; // шашка(дамка)

cin >> piece;

while (piece != 'm' && piece != 'k') {

cout << endl;

cout << "Error! You must input only m or k" << endl;

cout << "Piece (m or k) - ";

cin >> piece;

}

cout << "Input field's letter (a-h) - ";

cin >> letter;

while (letter != 'a' && letter != 'b' && letter != 'c' && letter != 'd' && letter != 'e' && letter != 'f' && letter != 'g' && letter != 'h')

{

cout << endl;

cout << "Error! You must input a letter only from a to h" << endl;

cout << "Input field's letter (a-h) - ";

cin >> letter;

}

cout << "Input field's number (1-8) - ";

cin >> number;

while (number < 1 && number >8)

{

cout << endl;

cout << "Error! You must input a number only from 1 to 8" << endl;

cout << "Input field's number (1-8) - ";

cin >> number;

}

Men men;

Kings kings;

Piece \*check1 = &men, \*check2 = &kings;

if (piece == 'm')

check1->Check(letter, number);

if (piece == 'k')

check2->Check(letter, number);

cout << endl;

cout << " Would you like to try again ('y' or 'n')? - ";

cin >> command;

while (command != 'y' && command != 'n')

{

cout << endl;

cout << "Error! You must input only y or n" << endl;

cout << endl;

cout << "Would you like to try again? - ";

cin >> command;

}

} while (command != 'n'); // ïîêè íå áóäå ââåäåíà êîìàíäà â³äì³íè

if (command == 'n')

exit(0);

return 0;

}