Завдання 1

Створити абстрактний клас «Figure» з методами обчислення площі та периметра, а також методом, що виводить інформацію про фігуру на екран. Створити похідні класи: «Rectangle» (прямокутник), «Circle» (коло), «Triangle» (трикутник) зі своїми методами обчислення площі та периметра. Створити масив n фігур і вивести повну інформацію про фігури на екран.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task1\_\_1\_

{

abstract class Figure

{

protected double P;

protected double S;

public abstract void Input();

public abstract void Show();

public abstract double Perimeter();

public abstract double Square();

}

class Rectangle : Figure

{

private double a;

private double b;

public override void Input()

{

Console.WriteLine("Введите стороны a и b прямоугольника:");

a = double.Parse(Console.ReadLine());

b = double.Parse(Console.ReadLine());

}

public override double Perimeter()

{

return P = 2 \* (a + b);

}

public override double Square()

{

return S = a \* b;

}

public override void Show()

{

Console.WriteLine("Стороны прямоугольника равны {0} и {1}, перимметр = {2}, площадь = {3}", a, b, P, S);

}

}

class Triangle : Figure

{

private double a;

private double b;

private double c;

public override void Input()

{

Console.WriteLine("Введите стороны a, b и c треугольника:");

a = double.Parse(Console.ReadLine());

b = double.Parse(Console.ReadLine());

c = double.Parse(Console.ReadLine());

}

public override double Perimeter()

{

return P = a + b + c;

}

public override double Square()

{

double p = (a + b + c) / 2;

return S = Math.Sqrt(p + (p - a) \* (p - b) \* (p - c));

}

public override void Show()

{

Console.WriteLine("Стороны треугольника равны {0}, {1}, {2}, перимметр = {3}, площадь = {4}", a, b, c, P, S);

}

}

class Circle : Figure

{

private double R;

public override void Input()

{

Console.WriteLine("Введите радиус круга R:");

R = double.Parse(Console.ReadLine());

}

public override double Perimeter()

{

return P = 2 \* Math.PI \* R;

}

public override double Square()

{

return S = Math.PI \* Math.Pow(R, 2);

}

public override void Show()

{

Console.WriteLine("Радиус круга равен {0}, перимметр = {1}, площадь = {2}", R, P, S);

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Доступные типы фигур:");

Console.WriteLine("1 - прямоугольник;\n2 - треугольник;\n3 - круг;");

int n;

Console.WriteLine("\nВведите количество фигур n:");

n = int.Parse(Console.ReadLine());

Figure[] Object = new Figure[n];

int type;

for (int i = 0; i < n; i++)

{

Console.WriteLine("\nВыберете тип фигуры:");

type = int.Parse(Console.ReadLine());

switch (type)

{

case 1:

Object[i] = new Rectangle();

Object[i].Input();

Object[i].Perimeter();

Object[i].Square();

Object[i].Show();

break;

case 2:

Object[i] = new Triangle();

Object[i].Input();

Object[i].Perimeter();

Object[i].Square();

Object[i].Show();

break;

case 3:

Object[i] = new Circle();

Object[i].Input();

Object[i].Perimeter();

Object[i].Square();

Object[i].Show();

break;

default:

Console.WriteLine("ERROR! Недопустимый вариант.");

break;

}

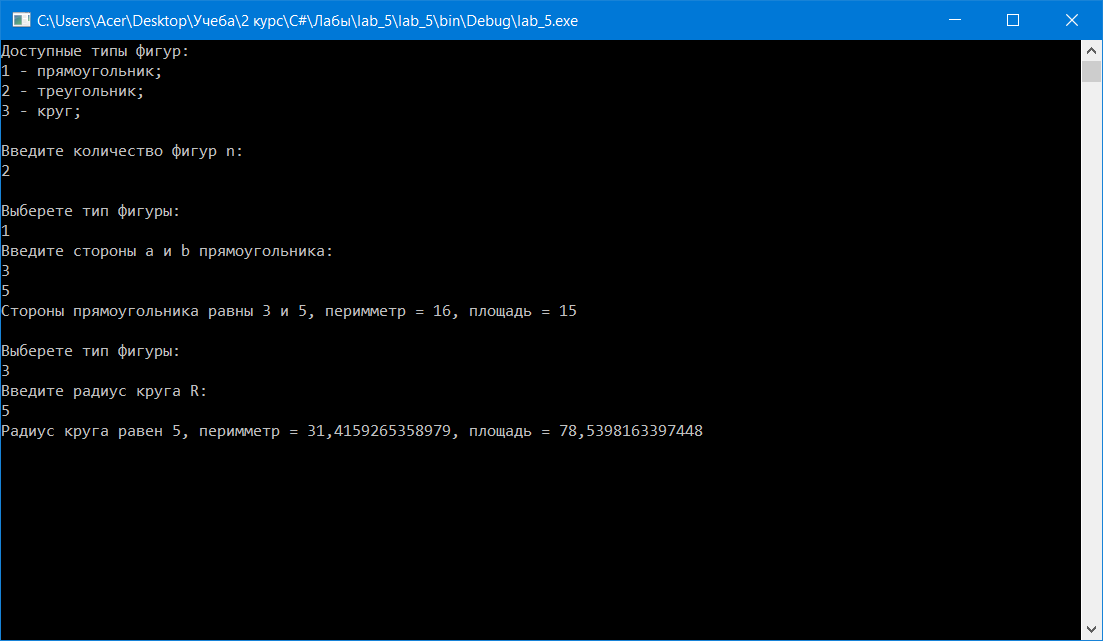
Console.ReadKey();

}

}

}

}



Задание 2

Описати клас «Книга в бібліотеці». Клас повинен включати в себе наступні поля (властивості):

* Прізвище та ім’я автора.
* Назва книги.
* Код.
* Рік видання.
* Кількість сторінок.
* Жанр.

Також необхідно:

* Визначити конструктор без параметрів, конструктори з параметрами.
* Продумати, які властивості повинні бути доступні тільки для перегляду, а які - і для зміни/установки значення.
* Визначити методи для порівняння двох книг за вказаним параметром.
* Реалізувати пошук книги по коду.
* Обов’язково реалізувати перевірку та обробку всіх нештатних ситуацій: книги з таким номер не існує, книг взагалі немає в списку, в поле коде встановлюються знаки пунктуації та інше.
* Всі класи описати в окремих файлах.
* Перевантажити стандартний метод toString();, для виводу інформації про книгу на екран.
* Книг повинно бути не менше п’яти.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_5

{

class Book

{

private string authorName;

private string authorSurname;

private string bookName;

private string code;

private int year;

private int pageCount;

private string genre;

public Book() { } //конструктор без параметров

public Book(string a, string b, string c, string d, int e, int f, string g) //конструктор с параметрами

{

this.authorName = a;

this.authorSurname = b;

this.bookName = c;

this.code = d;

this.year = e;

this.pageCount = f;

this.genre = g;

}

//методы сравнения 2 книг

public int CompareByAuthor(Book A)

{

if (pageCount.CompareTo(A.pageCount) == 0)

{

Console.WriteLine("Книги '{0}' и '{1}' имеют одинаковое количество страниц - {2};", bookName, A.bookName, A.pageCount);

return 1;

}

else

{

Console.WriteLine("Книги '{0}' и {1}' имеют разное количество страниц", bookName, A.bookName);

if (pageCount > A.pageCount)

{

Console.WriteLine("В книге '{0}' страниц больше чем в книге '{1}'", bookName, A.bookName);

}

else

{

Console.WriteLine("В книге '{0}' страниц больше чем в книге '{1}'", A.bookName, bookName);

}

return 0;

}

}

public int CompareByGenre(Book A)

{

if (genre.CompareTo(A.genre) == 0)

{

Console.WriteLine("Книги '{0}' и '{1}' имеют одинаковый жанр - {2};", bookName, A.bookName, A.genre);

return 1;

}

else

{

Console.WriteLine("Книги '{0}' и '{1}' имеют разные жанры.", bookName, A.bookName);

return 0;

}

}

//поиск книги по коду

public Book SearchByCode(Book[] A, ref string desire\_code)

{

for (int i = 0; i < A.Length; i++)

{

if ((String.Equals(A[i].code, desire\_code)))

return A[i];

}

return null;

}

//перегрузка метода ToString

public override string ToString()

{

string info = String.Empty;

info += ("Название книги: " + bookName + ";\n");

info += ("Автор: " + authorName + " " + authorSurname + ";\n");

info += ("Код - " + code + ";\n");

info += ("Год издания: " + year + ";\n");

info += ("Количество страниц книги: " + pageCount + ";\n");

info += ("Жанр книги: " + genre + ";\n");

return info;

}

}

class Program

{

static void Main(string[] args)

{

const int n = 5;

Book[] book = new Book[n];

book[0] = new Book("Эрнест", "Хемингуэй", "Старик и море", "1234", 1952, 320, "Повесть");

book[1] = new Book("Стивен", "Кинг", "Зеленая миля", "1531", 1996, 384, "Роман");

book[2] = new Book("Эрих Мария", "Ремарк", "Три товарища", "2532", 1932, 448, "Роман");

book[3] = new Book("Чак", "Паланик", "Бойцовский клуб", "3431", 1996, 555, "Роман");

book[4] = new Book("Марио", "Пьюзо", "Крестный отец", "4228", 1988, 555, "Роман");

Console.WriteLine("Книги в наличии:\n");

for (int i = 0; i < book.Length; i++)

Console.WriteLine(book[i]);

Console.WriteLine("Поиск книги по коду:");

int flag1 = 1;

while (flag1 == 1)

{

string desire\_code;

Console.WriteLine("Введите код для поиска нужной книги:");

desire\_code = Console.ReadLine();

Book searchBook = new Book();

searchBook = searchBook.SearchByCode(book, ref desire\_code);

if (searchBook != null)

{

Console.WriteLine(searchBook);

}

else Console.WriteLine("Книг с указанным кодом ({0}) не найдено.", desire\_code);

Console.Write("Продолжать поиск по коду? (да, нет): ");

string answer1 = Console.ReadLine();

if (String.Equals(answer1, "нет"))

break;

}

Console.WriteLine("\nСравнение 2 книг параметру:");

int flag = 1;

while (flag == 1)

{

Console.Write("Введите критерий сравнениия (объем или жанр): ");

string choice = Console.ReadLine();

Console.WriteLine("Введите номера сравниваемых книг (1-5):");

int a = int.Parse(Console.ReadLine());

int b = int.Parse(Console.ReadLine());

switch (choice)

{

case "объем":

book[a - 1].CompareByAuthor(book[b - 1]);

break;

case "жанр":

book[a - 1].CompareByGenre(book[b - 1]);

break;

default:

Console.WriteLine("Недопустимый вариант ответа");

break;

}

Console.Write("Продолжить сравнение? (да, нет): ");

string answer = Console.ReadLine();

if (String.Equals(answer, "нет"))

break;

}

}

}

}

