**ЛАБОРАТОРНАЯ РАБОТА 3**

**ИССЛЕДОВАНИЕ ПОЛИМОРФИЗМА**

**Цель лабораторной работы -** исследовать применения полиморфизма в языках объектно-ориентированного программирования на примере языка С++

**Задача.**

1. Написать программу на языке С++ для исследования полиморфизма согласно Варианту задания (табл. 10).
2. Исследовать виртуальные функции (табл. 11).

Таблица 10

**Варианты заданий**

| *№*  *Варианта* | *Описание классов* | *Задача* | |
| --- | --- | --- | --- |
|  | Базовый класс *Строка*: виртуальные функции вычисления длины и сортировки с выводом результата.  Производный класс *Большие буквы*: строка, конструктор с параметром, виртуальные функции вычисления длины и сортировки по возрастанию с выводом результата.  Производный класс *Маленькие буквы*: строка, конструктор с параметром, виртуальные функции вычисления длины и сортировки по убыванию с выводом результата. | Создать объекты наследуемых классов, вычислить длину и вывести строки из объектов, употребляя полиморфизм (указатель на объект базового класса) | |
|  | Базовый класс *Фигура*: виртуальные функции вычисления площади и периметра с выводом результата.  Производный класс *Треугольник*: координаты вершин, конструктор с параметрами, виртуальные функции вычисления площади и периметра с выводом результата.  Производный класс *Окружность*: радиус, конструктор с параметром, виртуальные функции вычисления площади и периметра с выводом результата. | Создать объекты наследуемых классов, вычислить и вывести площадь и периметр этих объектов, употребляя полиморфизм (указатель на объект базового класса) | |
|  | Базовый класс *Строка*: виртуальные функции вычисления длины и удаления символа с выводом результата.  Производный класс *Цифры*: строка, конструктор с параметром, виртуальные функции вычисления длины и удаления символа ‘5' с выводом результата.  Производный класс Буквы: строка, конструктор с параметром, виртуальные функции вычисления длины и удаления символа ‘a' с выводом результата. | Создать объекты наследуемых классов, вычислить длину и вывести строки из объектов, употребляя полиморфизм (указатель на объект базового класса) | |
|  | Базовый класс *Строка*: виртуальные функции вычисления длины и вставки символа с выводом результата.  Производный класс *Большие буквы*: строка, конструктор с параметром, виртуальные функции вычисления длины и вставки символа в указанную позицию с выводом результата.  Производный класс *Маленькие буквы*: строка, конструктор с параметром, виртуальные функции вычисления длины и вставки символа в указанную позицию с выводом результата. | Создать объекты наследуемых классов, вычислить длину и вывести отдельные строки объектов, употребляя полиморфизм (указатель на объект базового класса) | |
|  | Базовый класс *Фигура*: виртуальные функции вычисления площади и длины с выводом результата.  Производный класс *Эллипс*: полу оси, конструктор с параметрами, виртуальные функции вычисления площади и длины с выводом результата.  Производный класс *Окружность*: радиус, конструктор с параметром, виртуальные функции вычисления площади и длины с выводом результата. | Создать объекты наследуемых классов, вычислить и вывести площадь и периметр этих объектов, употребляя полиморфизм (указатель на объект базового класса) | |

Продолжение табл. 10

| *№*  *Варианта* | *Описание классов* | *Задача* | |
| --- | --- | --- | --- |
|  | Базовый класс *Строка*: виртуальные функции вычисления длины и замены символа с выводом результата.  Производный класс *Символы*: строка, конструктор с параметром, виртуальные функции вычисления длины и замены символа ‘#' на символы ‘!!’ с выводом результата.  Производный класс *Цифры*: строка, конструктор с параметром, виртуальные функции вычисления длины и замены символа ‘3' на символы ‘11' с выводом результата. | Создать объекты наследуемых классов, вычислить длину и вывести обработанные строки объектов, употребляя полиморфизм (указатель на объект базового класса) | |
|  | Базовый класс *Строка*: виртуальные функции вычисления длины и обработки строки с выводом результата.  Производный класс *Буквы*: строка, конструктор с параметром, виртуальные функции вычисления длины и обращения (вывод наоборот с конца в начало) строки с выводом результата.  Производный класс *Цифры*: строка, конструктор с параметром, виртуальные функции вычисления длины и увеличения строки в два раза с выводом результата. | Создать объекты наследуемых классов, вычислить длину и вывести обработанные строки объектов, употребляя полиморфизм (указатель на объект базового класса) | |
|  | Базовый класс *Фигура*: виртуальные функции вычисления площади и периметра с выводом результата.  Производный класс *Прямоугольник*: координаты вершин, конструктор с параметрами, виртуальные функции вычисления площади и периметра с выводом результата.  Производный класс *Окружность*: радиус, конструктор с параметром, виртуальные функции вычисления площади и периметра с выводом результата. | Создать объекты наследуемых классов, вычислить и вывести площадь и периметр этих объектов, употребляя полиморфизм (указатель на объект базового класса) | |
|  | Базовый класс *Строка*: виртуальные функции вычисления длины и количества символа с выводом результата.  Производный класс *Символы*: строка, конструктор с параметром, виртуальные функции вычисления длины и количества символа ‘\*’ с выводом результата.  Производный класс Буквы: строка, конструктор с параметром, виртуальные функции вычисления длины и количества символа ‘B' с выводом результата. | Создать объекты наследуемых классов, вычислить длину и вывести обработанные строки объектов, употребляя полиморфизм (указатель на объект базового класса) | |
|  | Базовый класс *Строка*: виртуальные функции вычисления длины и сдвига символов с выводом результата.  Производный класс *Цифры*: строка, конструктор с параметром, виртуальные функции вычисления длины и сдвига последнего символа на первое место с выводом результата.  Производный класс Буквы: строка, конструктор с параметром, виртуальные функции вычисления длины и сдвига первого символа на последнее место с выводом результата. | Создать объекты наследуемых классов, вычислить длину и вывести обработанные строки объектов, употребляя полиморфизм (указатель на объект базового класса) | |
|  | Базовый класс *Фигура*: виртуальные функции вычисления площади и периметра с выводом результата.  Производный класс *Квадрат*: координаты вершин, конструктор с параметрами, виртуальные функции вычисления площади и периметра с выводом результата.  Производный класс *Окружность*: радиус, конструктор с параметром, виртуальные функции вычисления площади и периметра с выводом результата. | Создать объекты наследуемых классов, вычислить и вывести площадь и периметр этих объектов, употребляя полиморфизм (указатель на объект базового класса) | |

Окончание табл. 10

| *№*  *Варианта* | *Описание классов* | | *Задача* |
| --- | --- | --- | --- |
|  | Базовый класс *Строка*: виртуальные функции вычисления длины и увеличения строки с выводом результата.  Производный класс *Буквы*: строка, конструктор с параметром, виртуальные функции вычисления длины и увеличения строки удвоением каждого символа с выводом результата (т.е. строка abc становиться aabbcc).  Производный класс *Символы*: строка, конструктор с параметром, виртуальные функции вычисления длины и увеличения строки удвоением символа ’k' с выводом результата (т.е. строка abcdkl будет .abcdkkl) | Создать объекты наследуемых классов, вычислить длину и вывести обработанные строки объектов, употребляя полиморфизм (указатель на объект базового класса) | |
|  | Базовый класс *Фигура*: виртуальные функции вычисления площади и периметра с выводом результата.  Производный класс *Трапеция*: координаты вершин, конструктор с параметрами, виртуальные функции вычисления площади и периметра с выводом результата.  Производный класс *Окружность*: радиус, конструктор с параметром, виртуальные функции вычисления площади и периметра с выводом результата. | Создать объекты наследуемых классов, вычислить и вывести площадь и периметр этих объектов, употребляя полиморфизм (указатель на объект базового класса) | |
|  | Базовый класс *Строка*: виртуальные функции вычисления длины и уменьшения строки с выводом результата.  Производный класс *Символы*: строка, конструктор с параметром, виртуальные функции вычисления длины и удаления символа, который стоит на нечетном месте с выводом результата.  Производный класс *Цифры*: строка, конструктор с параметром, виртуальные функции вычисления длины и удаления символа, который стоит на четном месте с выводом результата. | Создать объекты наследуемых классов, вычислить длину и вывести обработанные строки объектов, употребляя полиморфизм (указатель на объект базового класса) | |
|  | Базовый класс *Фигура*: виртуальные функции вычисления площади и периметра с выводом результата.  Производный класс *Ромб*: координаты вершин, конструктор с параметром, виртуальные функции вычисления площади и периметра с выводом результата.  Производный класс *Окружность*: радиус, конструктор с параметром, виртуальные функции вычисления площади и периметра с выводом результата. | Создать объекты наследуемых классов, вычислить и вывести площадь и периметр этих объектов, употребляя полиморфизм (указатель на объект базового класса) | |

**Теоретические сведения**

**Описание виртуальной функции**:

<описание виртуальной функции> ::= virtual <описание функции>

<описание чисто виртуальной функции> ::= virtual <описание прототипа функции> = 0;

**Использование виртуальной функции**:

<использование виртуальной функции> ::= <использование метода класса>

**Методические указания**

Если функция была объявленная как виртуальная, то она хранит это свойство во всех переопределениях в производных классах. В производном классе можно не использовать модификатор virtual.

Если в классе описанная чисто виртуальная функция, то такой класс называется абстрактным и объект данного класса не может быть создан.

Таблица 11

| Полиморфизм | | |
| --- | --- | --- |
| Виртуальные функции | | |
| Описание виртуальных функций | Описание чисто виртуальной функции | Использование виртуальных функций |

**Пример программы**

#include <stdio.h>

// interface for the CFigure class.

// Секция описания базового абстрактного класса представляющего

// геометрическую фигуру

//////////////////////////////////////////////////////////////////////

//константа числа Пи

#define PI 3.1415926535897932384626433832795

class CFigure

{

public:

//абстактные методы у них нет реализации поэтому объекты данного класса

//создавать невозможно

virtual void CalculateArea() =0; //чистая виртуальная функция

virtual void CalculateLength() =0; //чистая виртуальная функция

};

// implementation of the CFigure class.

// Секция реализации базового абстрактного класса отсутсвует

// Для него компилятор генерирует автоматически только конструктор

// и деструктор

//////////////////////////////////////////////////////////////////////

// interface for the CCircle class.

// Секция описания наследуемого класса окружность

//////////////////////////////////////////////////////////////////////

class CCircle : public CFigure

{

public:

CCircle(int rad); //констркутор с параметром радиус

//переопределяем виртуальные методы делаем класс не абстрактным

virtual void CalculateArea(); //метод для вычисления площади

virtual void CalculateLength(); //метод для вычисления длины

private:

int radius;

};

// implementation of the CCircle class.

// Секция реализации наследуемого класса окружность

//////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////

// Construction/Destruction

//////////////////////////////////////////////////////////////////////

//констркутор с параметром радиус

CCircle::CCircle(int rad):radius(rad)

{

}

//метод для вычисления площади

void CCircle::CalculateArea()

{

double area=PI\*radius\*radius;

printf( "\nCircle Area:\t%f %.2f", area, area);

}

//метод для вычисления длины

void CCircle::CalculateLength()

{

double length=2\*PI\*radius;

printf( "\nCircle length:\t%f %.2f", length, length);

}

// interface for the CEllipse class.

// Секция описания наследуемого класса Эллипс

//////////////////////////////////////////////////////////////////////

class CEllipse : public CFigure

{

public:

CEllipse(int a1,int a2);//констркутор с параметрами полуоси эллипса

virtual void CalculateArea();//метод для вычисления площади

virtual void CalculateLength(); //метод для вычисления длины

private:

//данные-члены хранят полуоси эллипса

int axis1;

int axis2;

};

// implementation of the CEllipse class.

// Секция реализации наследуемого класса эллипс

//////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////

// Construction/Destruction

//////////////////////////////////////////////////////////////////////

//констркутор с параметрами полуоси эллипса

CEllipse::CEllipse(int a1,int a2):axis1(a1),axis2(a2)

{

}

//метод для вычисления площади

void CEllipse::CalculateArea()

{

double area=PI\*axis1\*axis2;

printf( "\nEllipse Area:\t%f %.2f", area, area);

}

//метод для вычисления длины

void CEllipse::CalculateLength()

{

double length=2\*PI\*((axis1+axis2)/2);

printf( "\nEllipse Length:\t%f %.2f", length, length);

}

// Главная программа.

//

int main()

{

//Создаем объект класса окружность с указателем на базовый класс

CFigure\* circle\_figure=new CCircle(10);

//вызываем виртуальные методы через указатель на базовый класс

circle\_figure->CalculateArea();

circle\_figure->CalculateLength();

delete circle\_figure; //удаляем объект из памяти

//Создаем объект класса эллипс с указателем на базовый класс

CFigure\* ellipse\_figure=new CEllipse(10,15);

//вызываем виртуальные методы через указатель на базовый класс

ellipse\_figure->CalculateArea();

ellipse\_figure->CalculateLength();

delete ellipse\_figure; //удаляем объект из памяти

return 0; //код возврата

}

**Контрольные вопросы**

1. В чему состоит сущность полиморфизма?
2. Приведите примеры применения полиморфизма.
3. Что такое виртуальная функция?
4. Какая функция называется чисто виртуальной?
5. Как используются виртуальные функции?