

Жизненный цикл программы или этапы решения задач

«Машина должна работать, человек – думать»

Подготовила Наумкова К.Н.

Цель:

- показать из каких этапов складывается решение задач на ПК

Задача:

- на примере программы вычисления длины и площади круга, рассмотреть ключевые моменты каждой фазы решения задачи

Ход исследования:

1. Используя справочную литературу выделить технологию подготовки и решения задач на ЭВМ.
2. Рассмотреть из чего складываются основные этапы.
3. На примере задачи о нахождении длины и площади окружности рассмотреть ключевые моменты решения задач:
 - постановка задачи;
 - построение математической модели;
 - составление блок схемы;
 - написание листинга программы с помощью системы программирования Borland C++ Builder;
 - тестирование программы.
4. Результат исследования – запуск программы на выполнение.

Основные этапы решения задач



Решение задач включает следующие *этапы*, часть из которых осуществляется без участия ЭВМ.

1. Постановка задач:

- сбор информации о задаче;
- формулировка условия задачи;
- определение конечных целей;
- описание данных.

2. Построение математической модели.

3. Построение алгоритма:

- выбор формы записи алгоритма (блок-схема, табличная и др.);
- запись алгоритма.

4. Программирование:

- выбор языка программирования;
- выбор способа представления данных;
- запись алгоритма на выбранном языке;
- выбор тестов и методов тестирования.

5. Тестирование:

- проверка работоспособности программы.

6. Отладка:

- анализ результатов тестирования;
- устранение ошибок, совершенствование программы.

7. Сопровождение программы:

- доработка программы для решения конкретных задач;
- составление документации к использованию.

Алгоритм - это последовательность действий, приводящих к решению поставленной задачи.

Программа — одна или несколько последовательностей связанных команд (инструкций), которые, будучи выполнены компьютером, реализуют определенную функцию или операцию.

Язык программирования — совокупность средств, предназначенная для описания алгоритмов, реализуемых в программах ЭВМ.

Тестирование устанавливает факт наличия ошибки.

Отладка выясняет её причину.

Пример:

- Постановка задачи:

Написать программу вычисления длины окружности и площади круга.

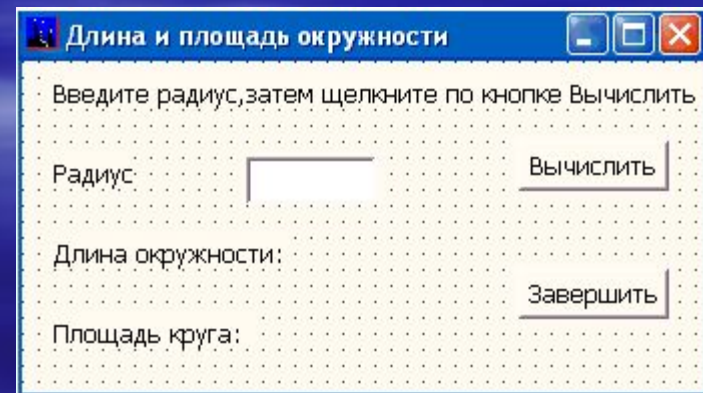
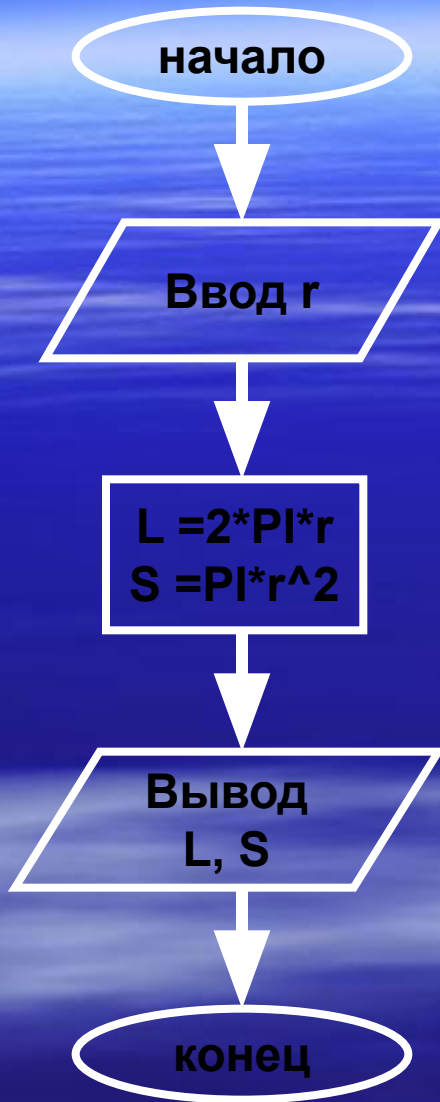
- r - радиус
 - L - длина окружности
 - S - площадь круга
 - радиус не может иметь отрицательное значение
- } дробные числа (float)

- Описание математической модели задачи:

- $L = 2 * \pi * r$
- $S = \pi * r^2$

Блок - схема

- Построение алгоритма в виде блок – схемы
- Программирование:
 1. Выбор языка программирования
Программа написана в интегрированной среде разработки приложений Borland C++ Builder, базирующаяся на объектно-ориентированном расширении языка C++
 2. Выбор способа представления данных



3. Листинг программы

```
#include <vcl.h>
#pragma hdrstop
#include <math.h>
#include "Radius_.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender) // щелчок по кнопке Вычислить
{
    float r;    // радиус
    float L;    // длина окружности
    float S;    // площадь круга
    if ( (Edit1->Text).Length() == 0) // проверим, введены ли данные в поле Радиус
    {
        MessageDlg ("Надо ввести радиус", mtInformation, TMsgDlgButtons() << mbOK, 0);
        if ( (Edit1->Text) .Length() == 0)
            Edit1->SetFocus();
        return;
    };
    r=StrToFloat (Edit1->Text); // получить данные из поля ввода
    L=2*M_PI*r; // вычислить длину окружности
    S=M_PI*r*r; // вычислить площадь круга
    // вывести результаты
    Label3->Caption="Длина окружности:" + FloatToStrF(L,ffGeneral,7,2);
    Label4->Caption="Площадь круга:" + FloatToStrF(S,ffGeneral,7,2);
}
```

Далее...

```

// нажатие клавиши в поле Радиус
void __fastcall TForm1::Edit1KeyPress(TObject *Sender, char &Key)
{
// коды запрещенных клавиш заменим нулем, в результате символы этих клавиш в поле
// редактирования не появятся
// Key – код нажатой клавиши
// проверим является ли символ допустимым
if ( (Key >='0') && (Key <= '9') ) // цифра
return;
// глобальная переменная DecimalSeparator содержит символ, используемый в качестве
// разделителя при записи дробных чисел
if (Key == DecimalSeparator)
{
if ( (Edit1->Text) . Pos (DecimalSeparator) !=0)
Key=0; // разделитель уже введен
return;
}
if (Key == VK_BACK) // клавиша <Backspace>
return;
if (Key == VK_RETURN) // клавиша <Enter>
{
Button1->SetFocus();
return;
};
// остальные клавиши запрещены
Key = 0; // не отображать символ
}
// щелчок по кнопке Завершить
void __fastcall TForm1::Button2Click(TObject *Sender)
{
Form1->Close(); // закрыть форму приложения
}

```

**«Чем реже встречаются ошибки, тем они ценнее»
Ежи Лец**

- **Тестирование:**
 - **проверка работоспособности программы**
Вычисление длины и площади окружности

Заключение:

1. Одна из основных задач, стоящих перед программистом, - добиться взаимопонимания с компьютером.
2. При выборе языка программирования необходимо учитывать следующие характеристики языка:
 - **Надежность** - язык должен обеспечивать минимум ошибок при написании программ. Более того, язык должен быть таким, чтобы неправильные программы было трудно писать.
 - **Удобочитаемость** - легкость восприятия программ человеком. Это характеристика важна при коллективной работе, когда несколько человек работают с одними и теми же текстами программ.
 - **Полнота** - характеризует способность описать класс задач в некоторой предметной области.
 - **Гибкость** - характеризует легкость выражения необходимых действий.

Библиография

- Голицина О.Л., Попова И.И. «Основы алгоритмизации и программирования»
- Соловьева Л.Ф. «Информатика в видеосюжетах»