

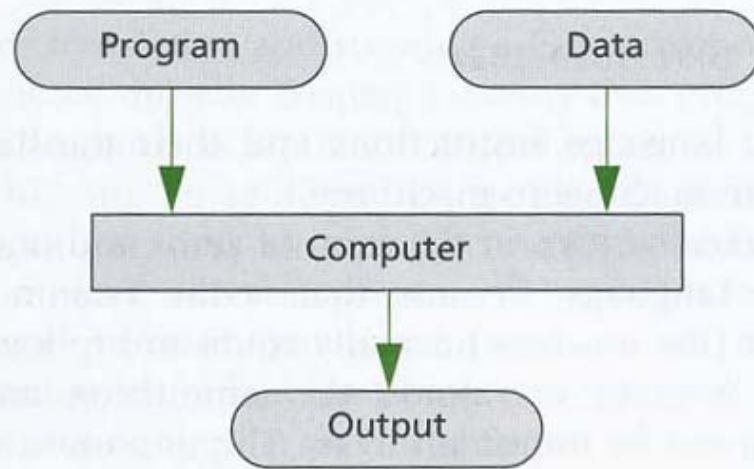
# Софтуер, Езици от високо ниво, Алгоритми, Типове данни

Базово програмиране  
Лекция 2  
Милена Карова  
Катедра КНТ ТУ-Варна



# Софтуер

## Операционна система



- Служи за интерфейс между потребител и машина
  - Присвоява компютърни ресурси на различните задачи, които компютърът трябва да изпълни
  - Управлява въвеждането, тестването и извеждането на данни

Популярни ОС: Unix, DOS, Windows, Linux, Mac OS, Android, iOS

**ПРОГРАМАТА** набор от инструкции, които компютърът изпълнява

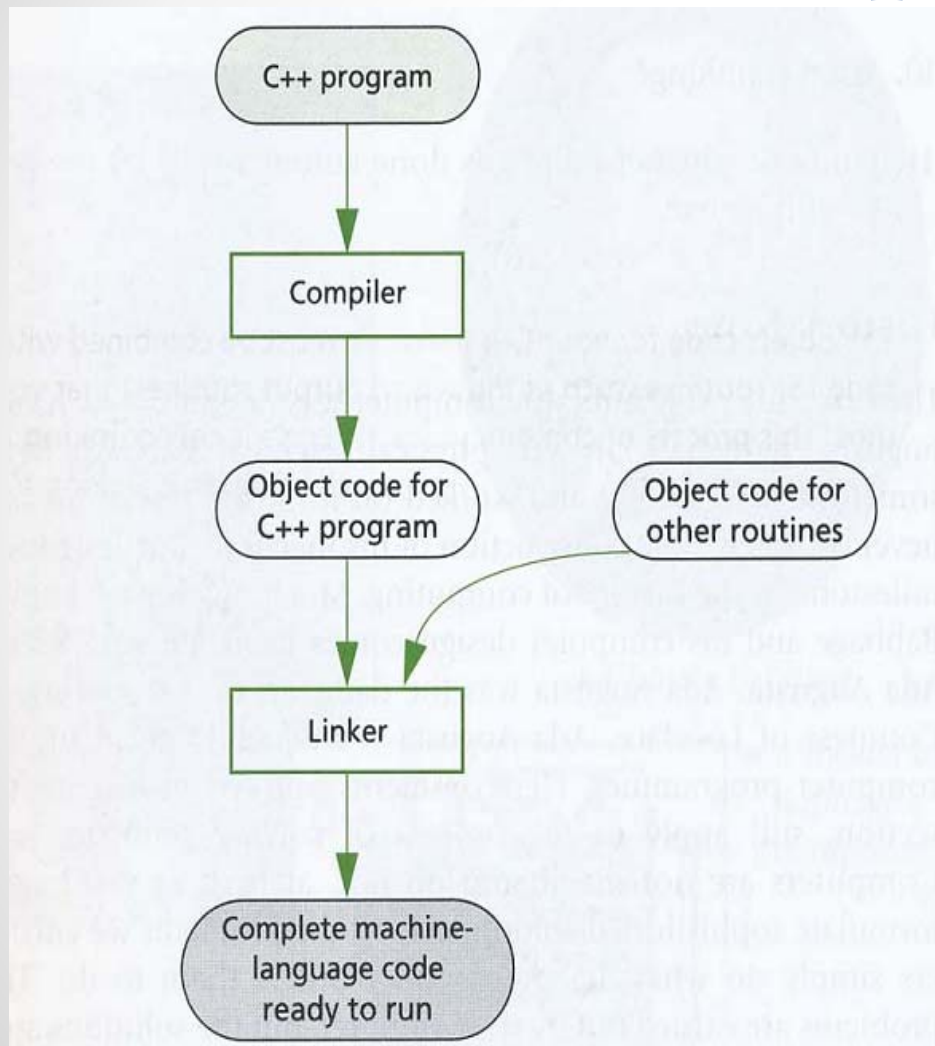
**ДАННИТЕ** са това което програмата обработва (най-често като вход) и извежда

Данните могат да се разглеждат като информация, подходяща и необходима за компютъра

# Езици от високо ниво

- Видове езици
  - Машинен език, бинарен (машинен) код
  - Асемблер, мнемонични кодове
  - Езици от високо ниво: Fortran, Pascal, Cobol, Lisp, Ada, Visual Basic, C/C++, Java, Perl, Python, PHP, HTML, ASP
  - **Пример за език от ниско ниво:** ADD X,Y,Z Събери числото от клетка X с това на клетка Y и запиши сумата в клетка Z.
  - **Преведена командата на машинен език:** 0110 1001 1010 1011
- Транслатори
  - Интерпретатори
  - Компилатори
- Компилаторът е програма, която превежда програма написана на език от високо ниво на машинен език, така че компютърът директно да разбере и изпълни програмата

# C++ Програма

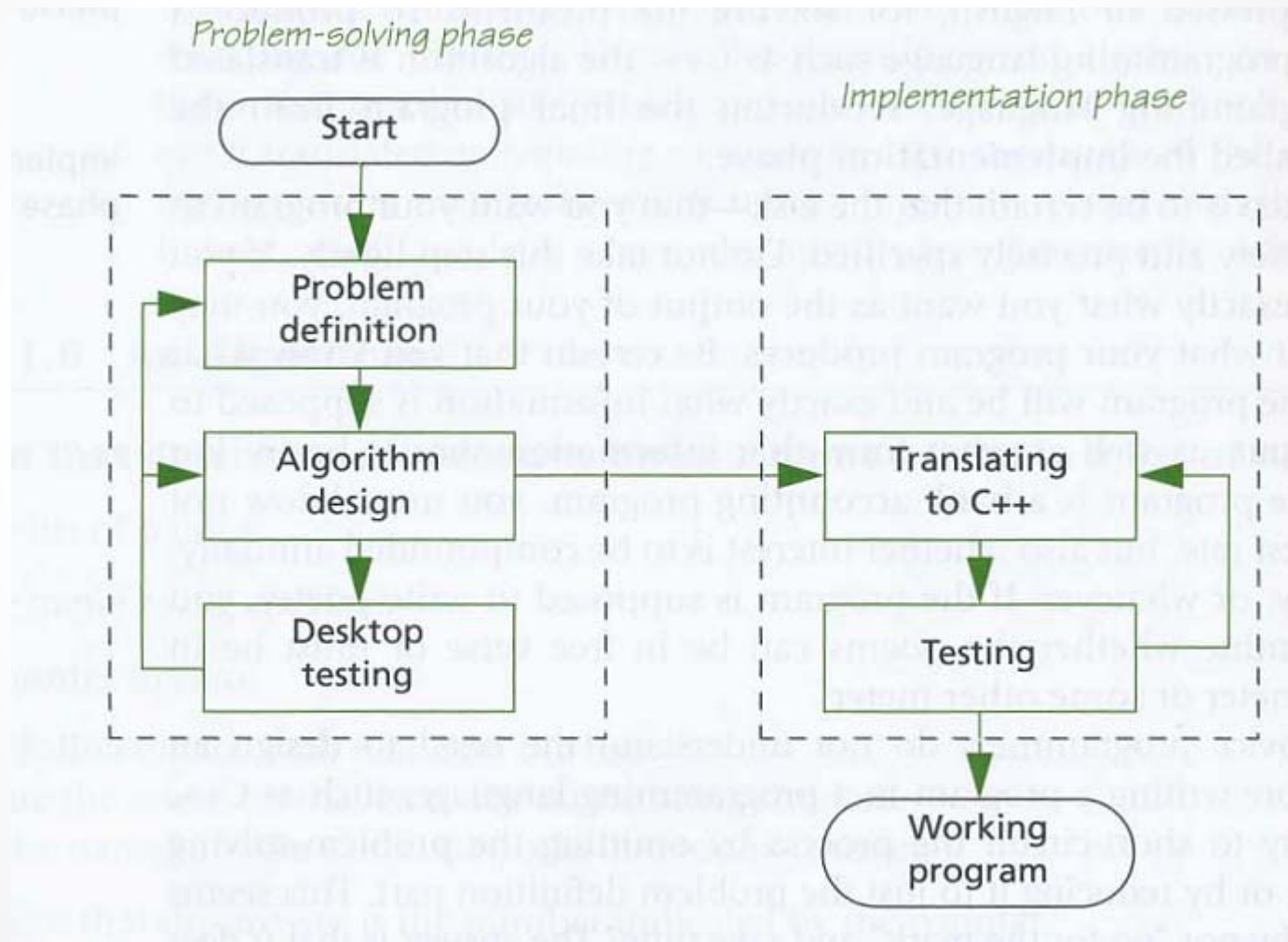


- Какво е source програма
- Какво е object програма
- Какво е code
- Какво е linker
- Обектният код трябва да се комбинира с обектният код на вече готови „действия“ (като различни типове на вход и изход на данните), които програмата използва. Процесът се нарича свързване (linking).

# Алгоритъм

- Откриване на метод за намиране на решение
- Методът лесно се програмира и се обръща в код
- Последователността от точни инструкции, които довеждат до решение на задачата се нарича алгоритъм
- Еквивалентни думи: рецепта, метод, насоки, процедура, програма
- Алгоритъмът се изразява чрез описание на разговорен език, блок-схеми и псевдокод
- Компютърната програма е пренаписан алгоритъм така че да се разбира и изпълнява от компютъра

# Проектиране на програма



# Проектиране на програма

- ООП- обектно ориентирано програмиране. Програмата се разглежда като сбор от взаимно действащи и влияещи се обекти.
- Програмиране с използване на ООП стил се свежда до проектиране на обектите и алгоритмите, които работят с тях.
- Algorithm design= Design of objects and their algorithms
- Основни характеристики на ООП: капсулиране, наследяване и полиморфизъм
- C++ използва класове при ООП

# Software life cycle (софтуерен цикъл)

1. Анализ и спецификация на задачите (problem definition)
2. Проектиране на софтуера (object and algorithm design)
3. Изпълнение (coding)
4. Тестване (Testing)
5. Поддръжка и развитие на системата (maintenance and evolution of the system)
6. Деинсталиране (Obsolescence)



# Въведение в C/C++

- Unix операционна система
- Появата на компилатори за други ОС
- Език от високо ниво с характерни черти на език от ниско ниво. Може директно да манипулира паметта на компютъра
- Добър език за писане на системни програми
- 1980 C се развива в C++ с цел по-лесна разбираемост на кода. Добавят се възможности за ООП език.

# Пример за C/C++ програма

```
#include <iostream>
using namespace std;
int main()
{ int broj, ed_cena,total;
  cout<<"Въведи брой закупени продукти\n";
  cin>>broj;
  cout<<endl<<"Въведи единична цена";
  cin>>ed_cena;
  total=broj*ed_cena;
  cout<<endl<<"Ако вие сте закупили ";
  cout<<broj<<" продукти и те струват "<< ed_cena<<" лева";
  cout<<endl<<" то вие сте платили "<< total<<" лева";
  return 0;
}
```

# Пример

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int number_of_pods, peas_per_pod, total_peas;
6     cout << "Press return after entering a number.\n";
7     cout << "Enter the number of pods:\n";
8     cin >> number_of_pods;
9     cout << "Enter the number of peas in a pod:\n";
10    cin >> peas_per_pod;
11    total_peas = number_of_pods * peas_per_pod;
12    cout << "If you have ";
13    cout << number_of_pods;
14    cout << " pea pods\n";
15    cout << "and ";
16    cout << peas_per_pod;
17    cout << " peas in each pod, then\n";
18    cout << "you have ";
19    cout << total_peas;
20    cout << " peas in all the pods.\n";
21    return 0;
22 }
```

```
Press return after entering a number.
Enter the number of pods:
10
Enter the number of peas in a pod:
9
If you have 10 pea pods
and 9 peas in each pod, then
you have 90 peas in all the pods.
```

# Анализ на програмата

- Деклариране на променливите
- Оператори
- `cin` и `cout` и пренасочване `>>` и `<<`
- Символ `*` и `\n` (разлика между `\n` и `endl`)
- `{` и `}`
- `#include <iostream>` директива
- `iostream` или `iostream.h` е хедърен файл и съдържа най-ваните входно-изходни функции (готови решения за вход и изход)
- Примери за хедърни файлове: `math`, `conio`, `stdio`, `string`.
- `using namespace std;`
- `int main()` или `void main()`
- `return 0;` или без `return`

# Етапи

- Етап на компилиране
- Етап на тестване и дебъгване
- Видове грешки: синтактични: съобщения и warning , грешки по време на изпълнение (run time error) и

# Константи

- Стойност, която по време на изпълнение на програмата не променя своята стойност
- Идентификаторът на константа е прието да се пише с главни букви
- `#define PI 3.14`
- `const double PI =3.14;`

# Променливи

- Идентификатор на променливата: `x`, `x1`, `x_1`, `_abc`, `ABC123z7`, `sum`, `rate`, `count`, `data2`, `Big_Boss`

Не се приемат `12`, `3x`, `%change`, `data-1`, `myfirst.c`, `prog.cpp`

С е case-sensitive език `RATE`, `rate`, `Rate`

Keywords или reserved words

- Деклариране на променливи

```
int number_of_bars;  
double one_weight, total_weight;
```

- Декларирането е задължително, защото е свързано с резервиране на клетки в оперативната памет

# Променливи

- Стойност на променливата:
  - Инициализиране `int count=5;` или `int count(5);` Стойността може да бъде константа или израз.
  - Неинициализирани променливи 0 и празен низ за глобални променливи

```
double rate = 0.07, time, balance = 0.0; double rate(0.07), time, balance(0.0);
```

- Присвояване `int count(5); count=count+2;`



# Типове данни

- Стандартни типове: int, char, float, double.
- Данните, които имат цяла и дробна част са реални и имат два различни записа:
  - Проста форма xxxxxx.xxxxxxxxxx
  - Експоненциална форма (scientific notation)

$3.67 \times 10^{17}$

3670000000000000000.0

3.67e17

$5.89 \times 10^{-6}$

0.00000589

5.89e-6

- В C++ са доразвити някои допълнителни типове: long int, long double,
- Long и Long Int са еднакви

`long big_total;`    `long int big_total;`

# Типове данни

Type Name	Memory Used	Size Range	Precision
<i>short</i> (also called <i>short int</i> )	2 bytes	-32,768 to 32,767	(not applicable)
<i>int</i>	4 bytes	-2,147,483,648 to 2,147,483,647	(not applicable)
<i>long</i> (also called <i>long int</i> )	4 bytes	-2,147,483,648 to 2,147,483,647	(not applicable)
<i>float</i>	4 bytes	approximately $10^{-38}$ to $10^{38}$	7 digits
<i>double</i>	8 bytes	approximately $10^{-308}$ to $10^{308}$	15 digits
<i>long double</i>	10 bytes	approximately $10^{-4932}$ to $10^{4932}$	19 digits

# Типът char

- Всеки символ: буква, цифра, пунктуационен знак
- char symbol, letter;
- ASCII код
- Типът string последователност от символи ( до 256)
- Разлика между тип char и тип string
- 'A' и "A"
- Типът string е специален клас в C++
- Използване на #include <string>

# Примери за използване на типа char и string

```
#include <iostream>
using namespace std;
int main()
{
    char symbol1, symbol2, symbol3;

    cout << "Enter two initials, without any periods:\n";
    cin >> symbol1 >> symbol2;

    cout << "The two initials are:\n";
    cout << symbol1 << symbol2 << endl;

    cout << "Once more with a space:\n";
    symbol3 = ' ';
    cout << symbol1 << symbol3 << symbol2 << endl;

    cout << "That's all.";

    return 0;
}
```

Enter two initials, without any periods:

J B

The two initials are:

JB

Once more with a space:

J B

That's all.

# Примери за използване на типа char и string

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string middle_name, pet_name;
    string alter_ego_name;

    cout << "Enter your middle name and the name of your pet.\n";
    cin >> middle_name;
    cin >> pet_name;

    alter_ego_name = pet_name + " " + middle_name;

    cout << "The name of your alter ego is ";
    cout << alter_ego_name << "." << endl;

    return 0;
}
```

```
Enter your middle name and the name of your pet.
Parker Pippen
The name of your alter ego is Pippen Parker.
```

```
Enter your middle name and the name of your pet.
Parker
Mr. Bojangles
The name of your alter ego is Mr. Parker.
```