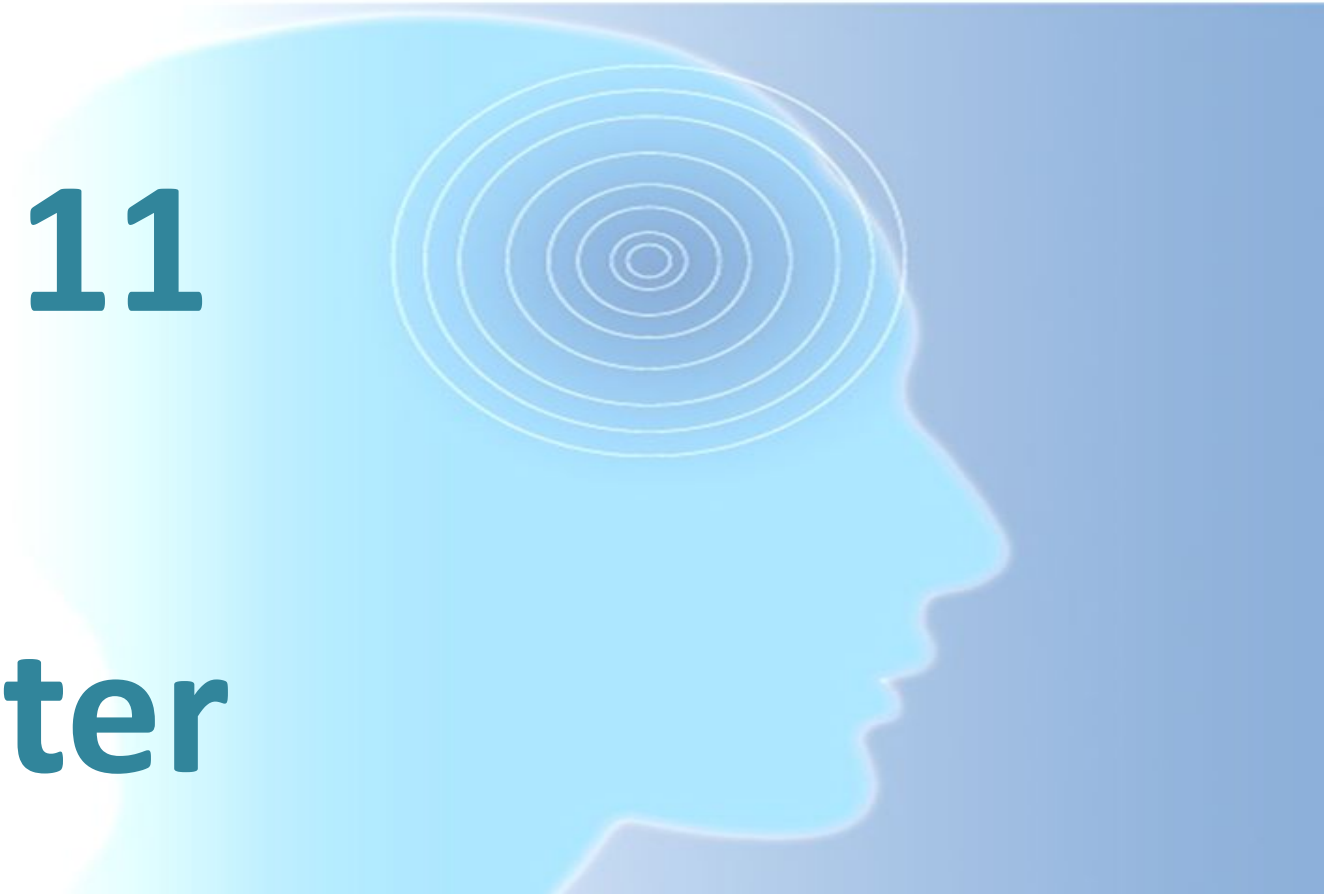
Revision of the material covered



Grade 11

1 quarter

UNIT 11.1A
COMPUTER SYSTEMS

UNIT 11.1B
PROGRAMMING PARADIGMS

UNIT 11.1C
SYSTEMS LIFECYCLE

Unit 11.1A Computer Systems

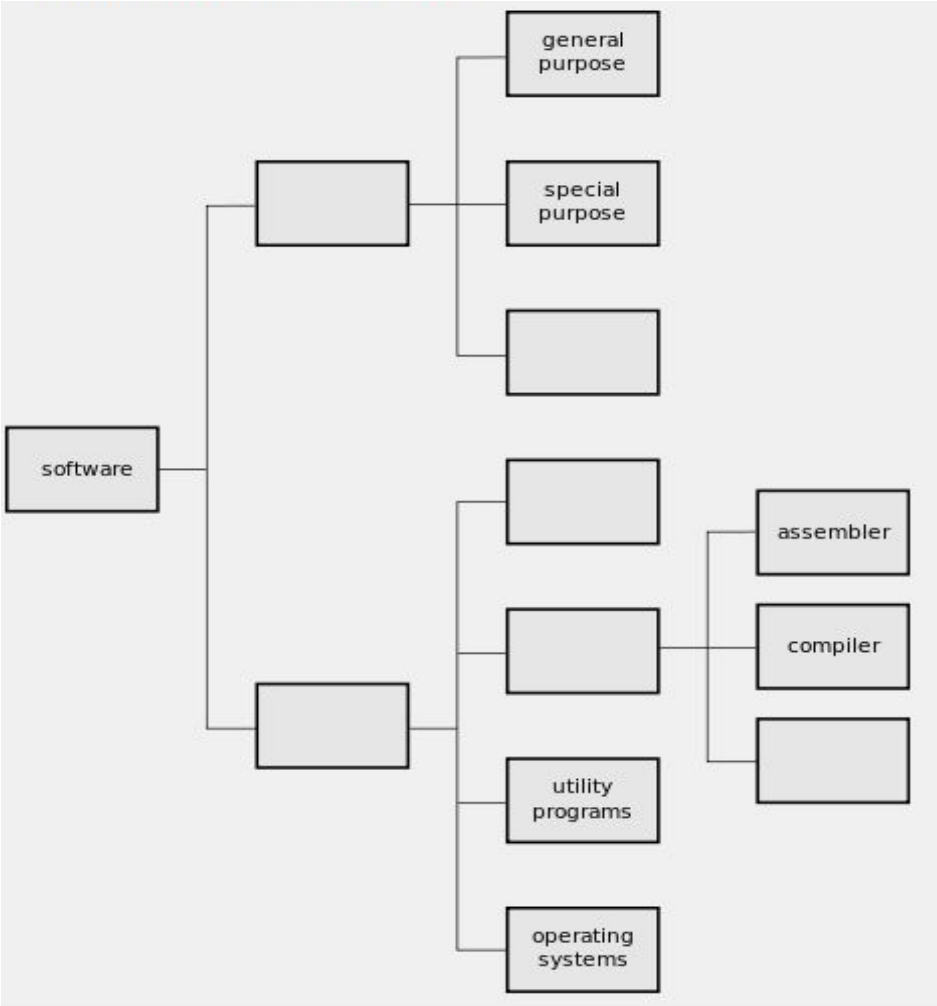
11.3.1.1 justify the choice of software and selection criteria for specific purposes
11.3.1.2 classify application software
11.3.1.3 describe the purpose and basic functions of operating systems
11.3.1.4 compare single-user and multi-user operating systems
11.3.1.5 compare single-tasking and multitasking operating systems
11.3.2.1. describe the interaction of the central processor with peripheral devices
11.3.2.2. describe the purpose of the components of the CPU, system bus and main memory.
11.3.4.1 explain the differences between RAM and ROM
11.3.4.2 explain the purpose of virtual memory
11.3.4.3 explain the purpose of the cache
11.3.3.1 to distinguish the laws of Boolean algebra
11.3.3.2 simplify logical expressions using the laws of Boolean algebra
11.3.3.3 build truth tables AND, OR, NOT, NAND, NOR, XOR

11.3.1.1 justify the choice of software and selection criteria for specific purposes

11.3.1.2 classify application software

1. Fill in the empty cells in this scheme

2. Give examples for each type of software.



Questions

Questions

1. Which one of the following is system software: word processor, disk formatter, database?
2. Give an example of a type of language translator.
3. Explain what an assembler is used for.
4. What type of language translator is required for a high-level programming language such as C#?
5. What is the code the compiler reads? What is the code the compiler creates?

Questions

6. Why is bespoke software more expensive to buy than off-the-shelf software?
7. Which of the following is general-purpose software and which is special-purpose software: a spreadsheet, an accounting package, a presentation package, a photo editor?
8. What is meant by application software?

Questions

9. Which generations of programming language are classed as low-level languages? Why?
10. Why are there so many high-level languages?
11. What is the relationship between high-level language code and low-level language code?

Answers

System software

Q1 Disk formatter is part of system software.
Q2 Types of language translators: assembler, interpreter, compiler.
Q3 An assembler is used to translate assembly code into machine code.
Q4 C# is translated using a compiler.
Q5 The compiler reads source code and produces object code.

Application software

Q6 Both types of software cost the same amount of money to develop. Bespoke software is paid for wholly by the client that requires this software. The development cost of off-the-shelf software is spread over a wide customer base, so is cheaper for an individual customer.
Q7 General purpose software: spreadsheet.
Special purpose software: accounting package, photo editor, presentation package.
Q8 Application software is a program or series of programs that allows a user to perform non- computer tasks.

Generations of programming languages

Q9 First and second generation languages are classed as low level as the instruction sets they use reflect the processor architecture.
Q10 There are many types of problems to be solved by computer and different languages were developed to make it easier to solve these problems.
Q11 One high-level language statement will generally be translated into several low-level language statements.

11.3.1.3 describe the purpose and basic functions of operating systems

11.3.1.4 compare single-user and multi-user operating systems

11.3.1.5 compare single-tasking and multitasking operating systems

What is an operating system?

What is the basic functions OS?

Name of types OS?

Describe and give examples for each type of OS?

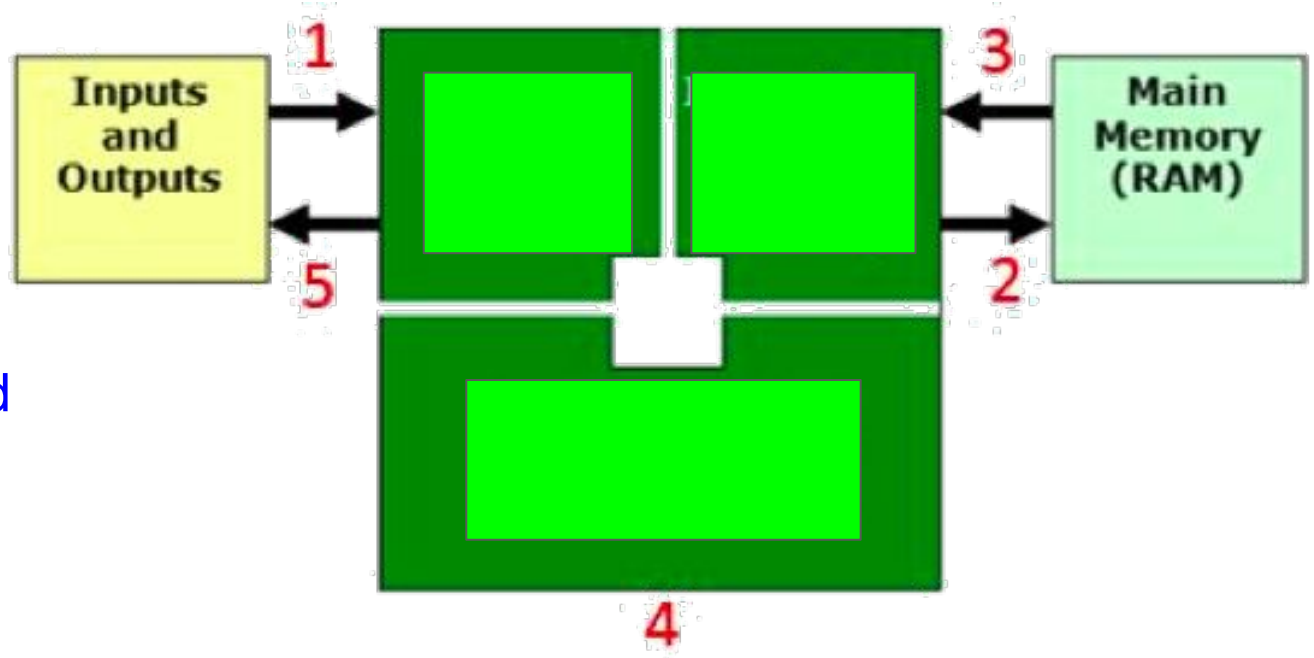
What is GUI and CLI? Where their could use?

11.3.2.1. describe the interaction of the central processor with peripheral devices
11.3.2.2. describe the purpose of the components of the CPU, system bus and main memory.

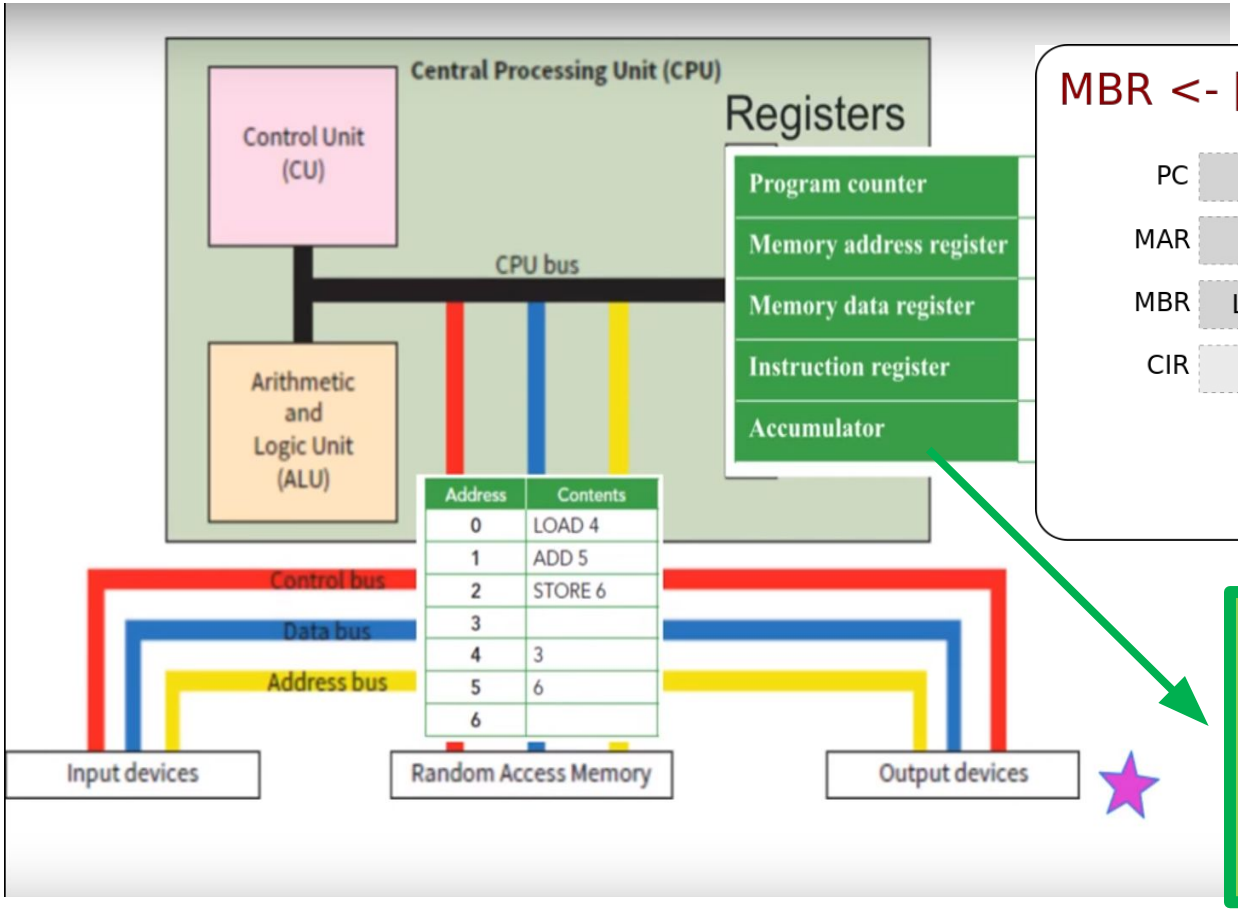
1. Describe processes in this scheme, and name of each components in the CPU.

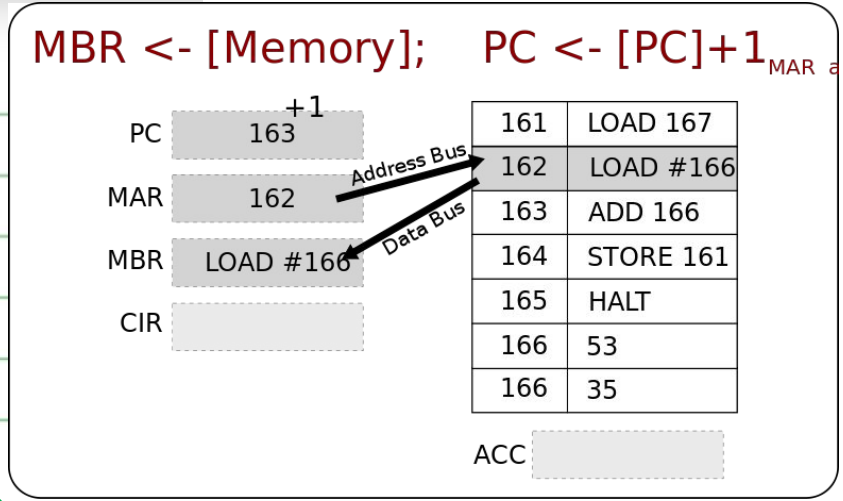
2. Call name black arrows in this scheme?

3. Explain of the CPU, system bus and main memory functions?



Describe the Fetch Execute Cycle using registers?



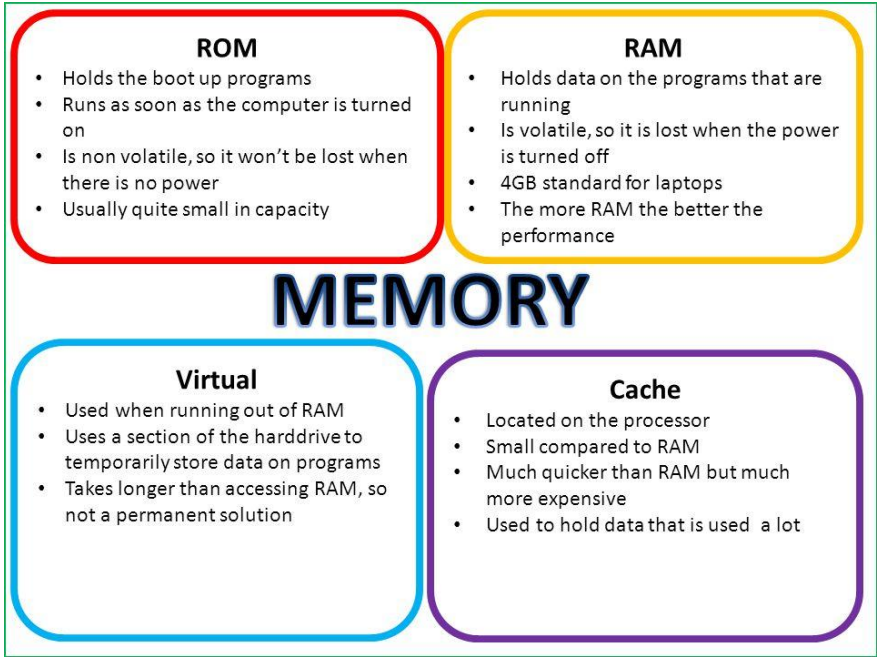


Program counter
Memory Address Register
Memory Data Register
Instruction register
Accumulator

11.3.4.1 explain the differences between RAM and ROM
11.3.4.2 explain the purpose of virtual memory
11.3.4.3 explain the purpose of the cache

Memory can be divided into two types: _____ and _____.

What is a RAM?
What is a ROM?
What is a Virtual Memory?
What is a Cache?



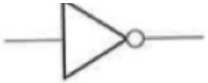
Fill in the missing words

You cannot write to _____, you can only read from it	_____ is a type of non-volatile memory
_____ allows you to both read and write data	There may be just a few megabytes of _____ in a computer.
_____ is used as main memory to hold both data and programs	_____ chips are located in removable memory modules that are slotted into sockets on the motherboard. This means they can be easily removed and updated.
_____ is used to hold basic computer hardware settings and in the past it held the BIOS to boot up the computer	_____ chips are usually located on the motherboard or printed circuit board such as a graphics card. They are not removable as they are soldered into the motherboard.
_____ is a type of volatile memory	There are usually gigabytes of _____ in a computer

11.3.3.1 to distinguish the laws of Boolean algebra

11.3.3.2 simplify logical expressions using the laws of Boolean algebra

11.3.3.3 build truth tables AND, OR, NOT, NAND, NOR, XOR



Inputs		Output
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

Inputs		Output
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Input	Output
A	X
0	1
1	0

AND, OR, NOT, NAND, NOR, XOR

Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Inputs		Output
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Laws of Boolean algebra

$$A \wedge 1 = A$$

$$A \vee 1 = 1$$

$$\neg A \wedge A = 0$$

$$A \vee 0 = A$$

$$A \wedge 0 = 0$$

$$\neg A \vee A = 1$$

Unit 11.1B Programming Paradigms

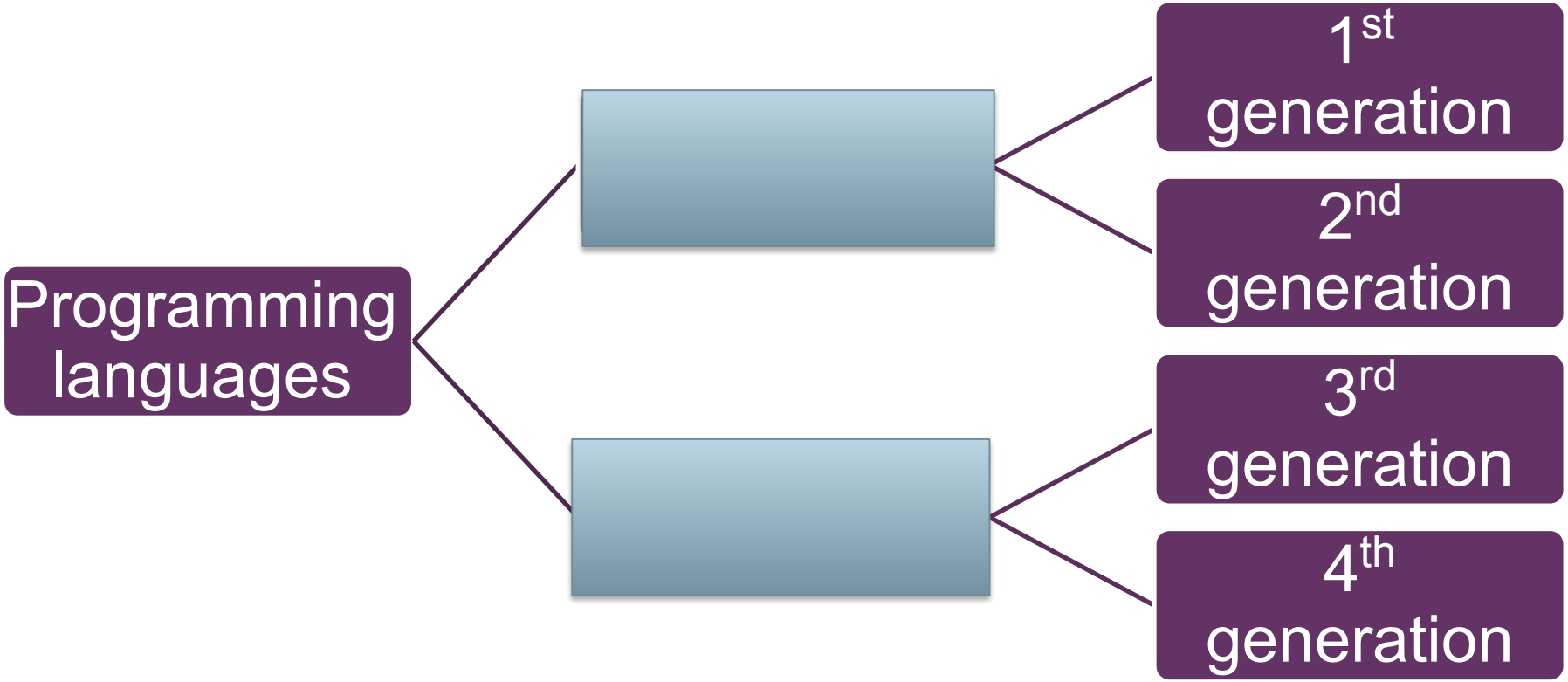
11.5.1.1 to distinguish between the generations of programming languages

11.5.1.2 classify low and high level programming languages

11.5.1.3 to analyze a simple program in assembler language

11.5.1.4 use trace tables to find and validate the algorithm

Programming language divide into 2 groups:



Compare two programming languages, and solve this task with Java program language.

Assembly Language Code

```
        INP          00 INP
        STA A        01 STA 08
        INP          02 INP
        STA B        03 STA 09
        LDA A        04 LDA 08
        ADD B        05 ADD 09
        OUT          06 OUT
        HLT          07 HLT
A       DAT          08 DAT 00
B       DAT          09 DAT 00
```

```
Program A1;
 var a, b, s: integer;
Begin
Write ('введите a');
Read (a);
Write ('введите b');
Read (b);
S:=a+b;
Write ('S=', S);
End.
```

Программа сложения двух чисел

C

```c
#include <stdio.h>

int main(void) {
   int a, b;
   scanf("%d", &a);
   scanf("%d", &b);
   printf("%d\n", a + b);
   return 0;
}
```

C++

```cpp
#include <iostream>

using namespace std;

int  main() {
   long a, b;
   cin >> a >> b;
   cout << a + b << endl;
   return 0;
}
```

Pascal

```pascal
var a, b: longint;
begin
   readln(a, b);
   writeln(a + b);
end.
```

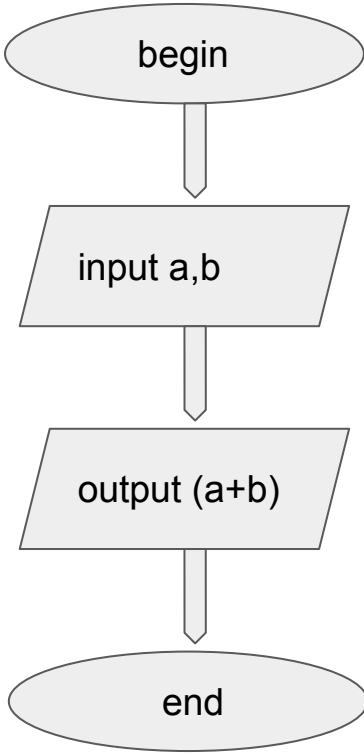
Java

```java
import java.util.Scanner;

class Main {
        public static void main(String args[]) {
                Scanner s = new Scanner(System.in);
                System.out.println(s.nextInt() +
s.nextInt());
        }
}
```

Pyton

```python
from string import split
a,b=map(int,split(raw_input()))
print a+b
```



Trace table

(b) Trace the assembly language program using the trace table.

```
300    LDD    321
301    INC
302    STO    323
303    LDI    307
304    INC
305    STO    322
306    END
307    320
```

```
320    49
321    36
322    0
323    0
```

Trace table:

Accumulator	Memory address			
	320	321	322	323
	49	36	0	0

[6]

Trace table

Let's see a trace table in action! For the following program, a trace table is created to determine the values of the variables in each step.

PHP | Java | C++ | C# | Visual Basic | Python

```php
<?php
  $x = 10;
  $y = 15;
  $z = $x * $y;
  $z++;
  echo $z;
?>
```

The trace table for this program is shown below. Notes are optional, but they help the reader to better understand what is really happening.

Step	Statement	Notes	$x	$y	$z
1	`$x = 10`	The value 10 is assigned to variable `$x`.	**10**	?	?
2	`$y = 15`	The value 15 is assigned to variable `$y`.	10	**15**	?
3	`$z = $x * $y`	The result of the product `$x * $y` is assigned to `$z`.	10	15	**150**
4	`$z++`	Variable `$z` is incremented by one.	10	15	**151**
5	`echo $z`	The value 151 is displayed.			

Translators

1) Name types of translators

2) What their roles are, and what are the differences between compilers and interpreters.

Unit 11.1C Systems Lifecycle

11.2.1.8 develop requirements for the new system based on the information collected

11.2.1.7 use flowcharts to represent input, processing, storage and output in computing systems

11.2.1.6 use data flow diagrams (DFD-1 level) to represent input, processing, storage and output in computing systems

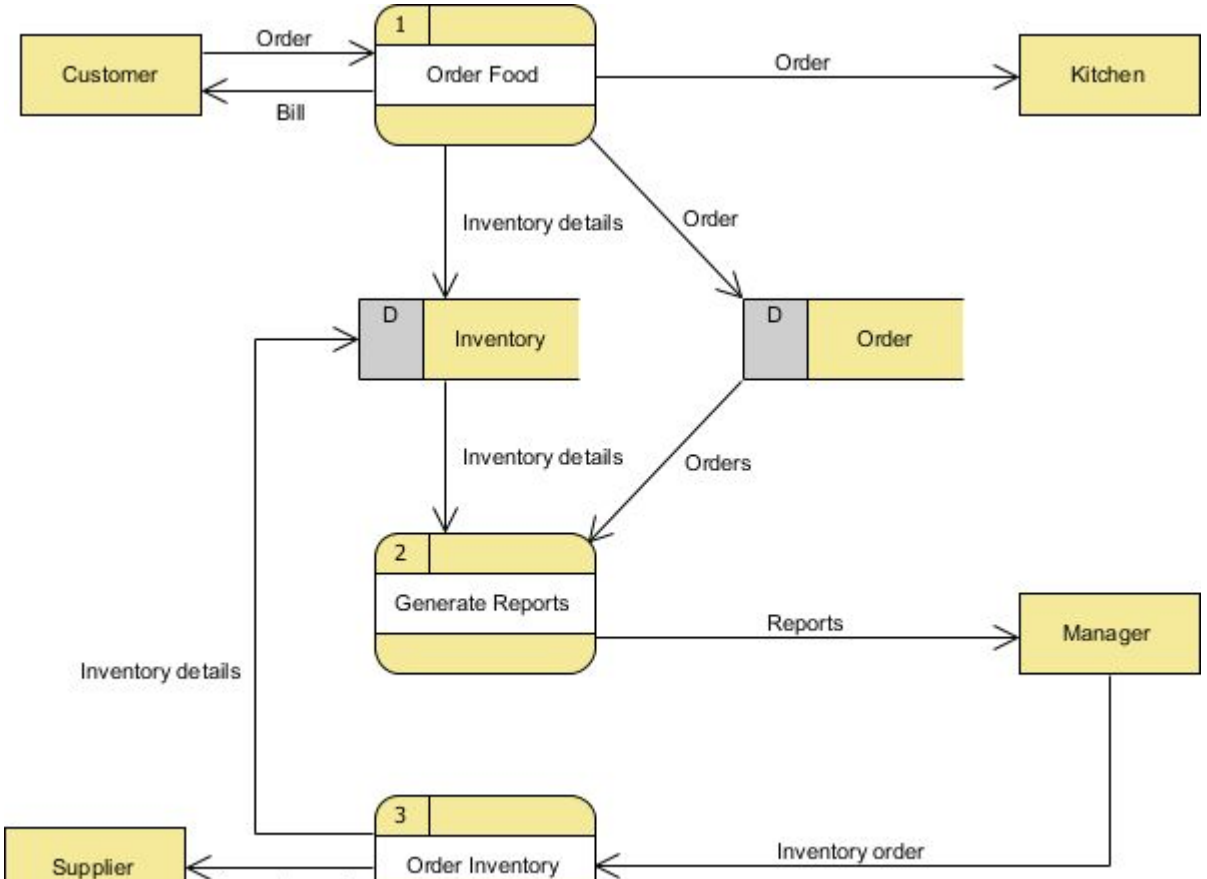
- What is a System Lifecycle?

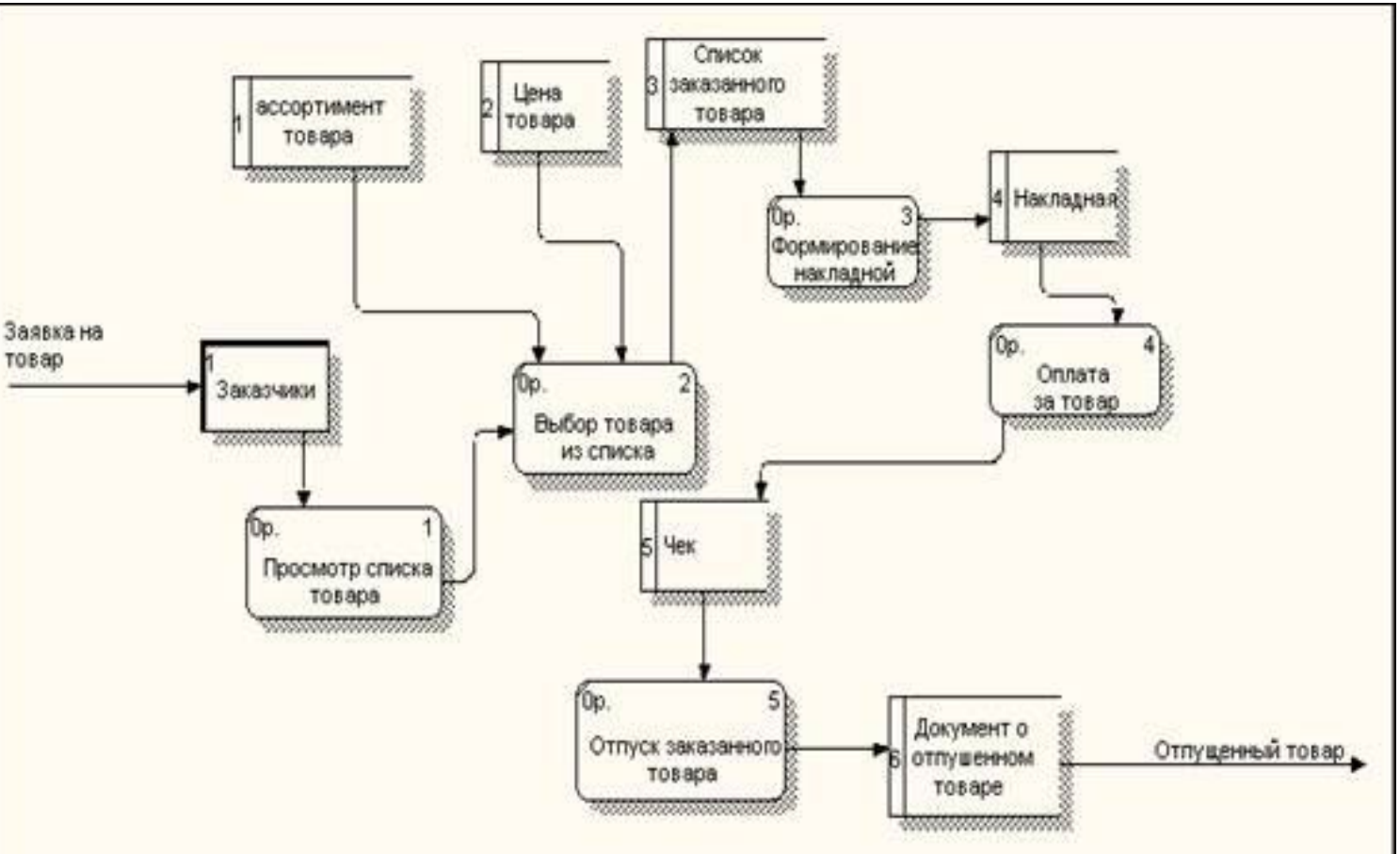
- Name the type of models SLC?

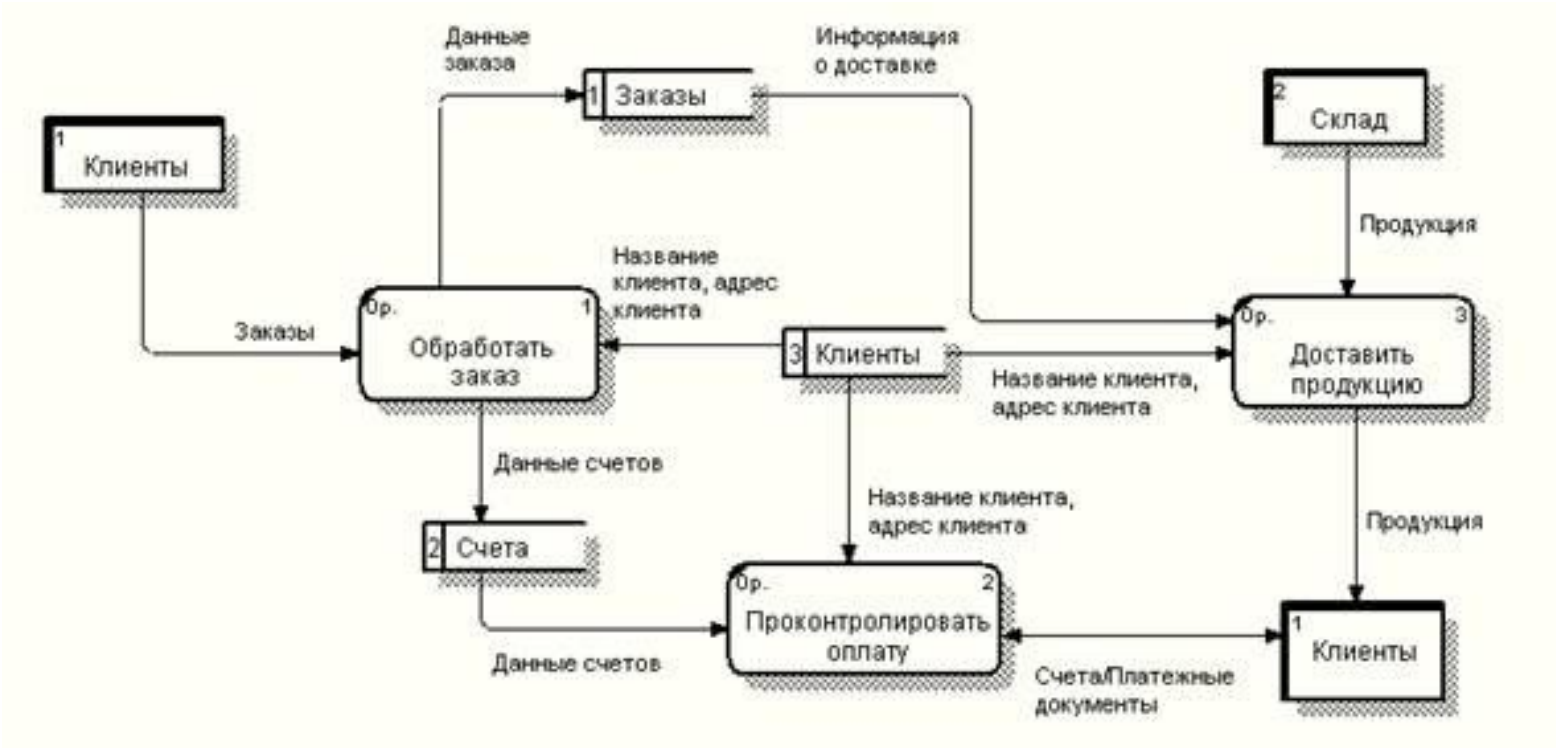
Questions

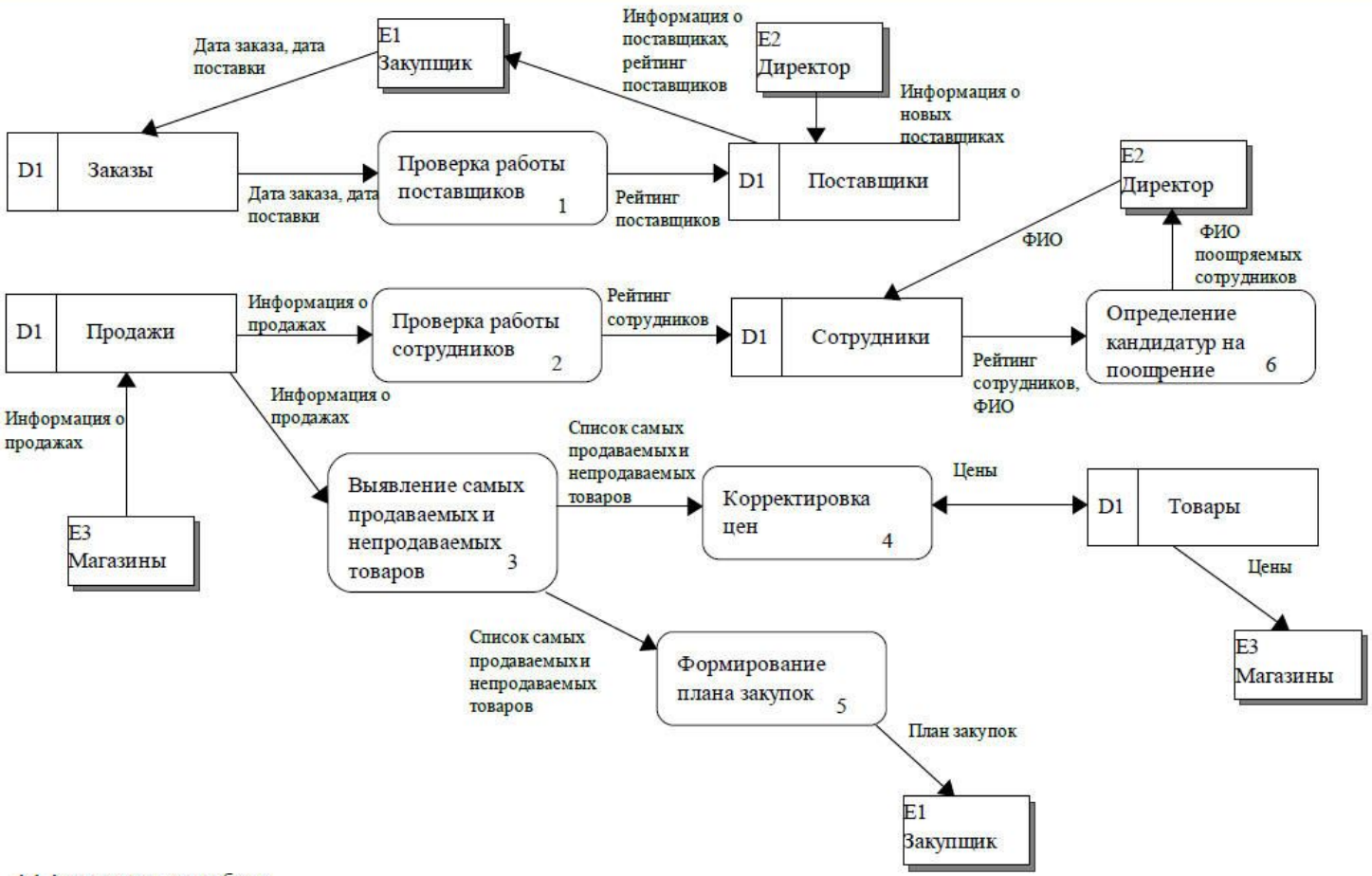
1. State the stages of the waterfall model for the systems life cycle.
2. What documents could you ask to look at?
3. What fact-finding techniques could be used when analysing this problem? Explain why each technique is used.
4. What is prototyping? State three reasons why prototyping is used.
5. What are the limitations of the waterfall model?
6. How does the spiral model differ from the waterfall model?

Where we can use DFD?
Based on the example make DFD pizza order




1 ассортимент товара
2 Цена товара
3 Список заказанного товара
4 Накладная
0р. 3 Формирование накладной
Заявка на товар
1 Заказчики
0р. 2 Выбор товара из списка
0р. 4 Оплата за товар
0р. 1 Просмотр списка товара
5 Чек
0р. 5 Отпуск заказанного товара
6 Документ о отпущенном товаре
Отпущенный товар


Данные заказа
Информация о доставке
1 Заказы
2 Склад
1 Клиенты
Продукция
Название клиента, адрес клиента
Заказы
0р. 1 Обработать заказ
3 Клиенты
0р. 3 Доставить продукцию
Название клиента, адрес клиента
Данные счетов
Название клиента, адрес клиента
Продукция
2 Счета
0р. 2 Проконтролировать оплату
1 Клиенты
Данные счетов
Счета/Платежные документы



A4 Анализировать работу