

Основи програмування

Бивойно Павло Георгійович

Параметри курсу

Комп'ютерна інженерія

| Форма навчан. | Рік навч. | Сем. | Розподіл годин | | | | | За тиждень | | ІНДЗ | Контр. |
|---------------|-----------|------|----------------|-----------|-----------|------------|------------|------------|-----|-----------|----------|
| | | | Всього ауд. | Лек | Лаб. | СРС | Разом | Ауд. | СРС | | |
| Денна КІ | 1 | 1 | 60 | 32 | 28 | 150 | 210 | 3.8 | 9.4 | | Е |
| | 1 | 2 | | | | 90 | 90 | | 6.4 | КП | |
| Усього | | | 60 | 32 | 28 | 240 | 300 | | | КП | Е |

Інженерія програмного забезпечення

| Форма навчан. | Рік навч. | Сем. | Розподіл годин | | | | | За тиждень | | ІНДЗ | Контр. |
|---------------|-----------|------|----------------|-----------|-----------|------------|------------|------------|-----|-----------|----------|
| | | | Всього ауд. | Лек | Лаб. | СРС | Разом | Ауд. | СРС | | |
| Денна ІПЗ | 1 | 1 | 60 | 32 | 28 | 150 | 210 | 3.8 | 9.4 | | Е |
| | 1 | 2 | | 20 | 20 | 110 | 90 | 2.9 | 7.9 | КП | |
| Усього | | | 60 | 52 | 48 | 260 | 360 | | | КП | Е |

Література

1. Шпак З.Я. Програмування мовою С / З.Я. Шпак. – Львів : вид-во НУ «Львівська політехніка», 2011. – 436 с.
2. Керниган Б., Ритчи Д. Язык программирования С / Брайан Керниган, Деннис Ритчи – М. : Вильямс, 2013. – 304 с.
3. Шилдт Г. Полный справочник по С / Г. Шилдт. – М. : Вильямс, 2009. – 704 с.
4. Kernighan B. W., Ritchie D. M. C Programming Language / Dennis M. Ritchie, Brian W. Kernighan, – 2nd ed. – Prentice-hall, inc., 1988. – 263 p.
5. Kochan S.G. Programming in C/ Stephen G. Kochan – 3rd ed. – Sams Publishing, 2004. – 505 p.

Історія розвитку ЕОМ

1941 Німеччина, Конрад Цузе,
електромеханічний (на реле)
програмований комп'ютер Z3

1946 Джон Моклі, США,
електронний (електронні лампи)
комп'ютер ENIAC

(Джон фон Нейман запропонував
зберігати програму у пам'яті)

Перша ЕОМ в Україні

4 січня 1952 р. Президія АН СРСР заслухала доповідь Сергія Лебедева про введення в експлуатацію малої електронно-цифрової обчислювальної машини "МЭСМ".

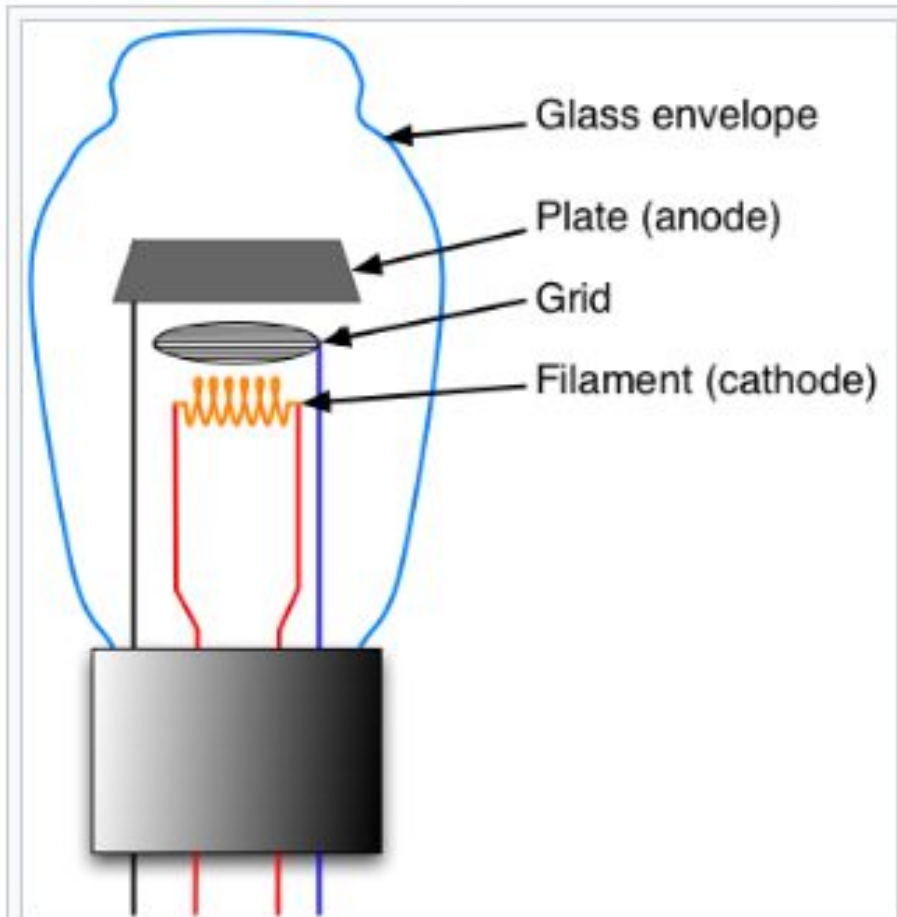
В 1952 р. "МЭСМ" була практично єдиною в країні ЕОМ, на якій проводились обчислення найважливіших науково-технічних задач в галузі термоядерних процесів, космічних польотів та ракетної техніки...

Як вони виглядали?

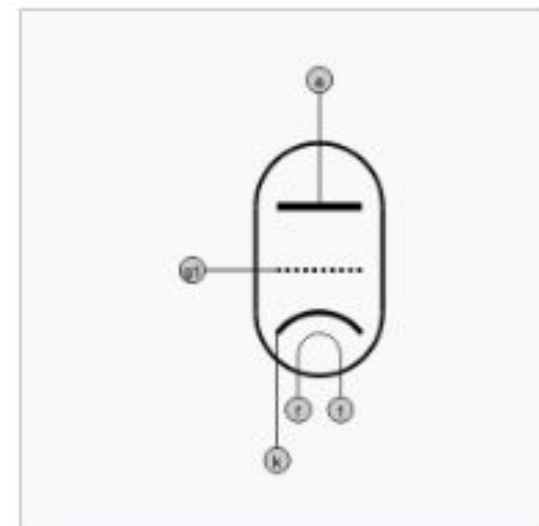
ENIAC важила 30 тон
MESM споживала 25 кВт,
Займала площу 60 кв.м.

Чому?

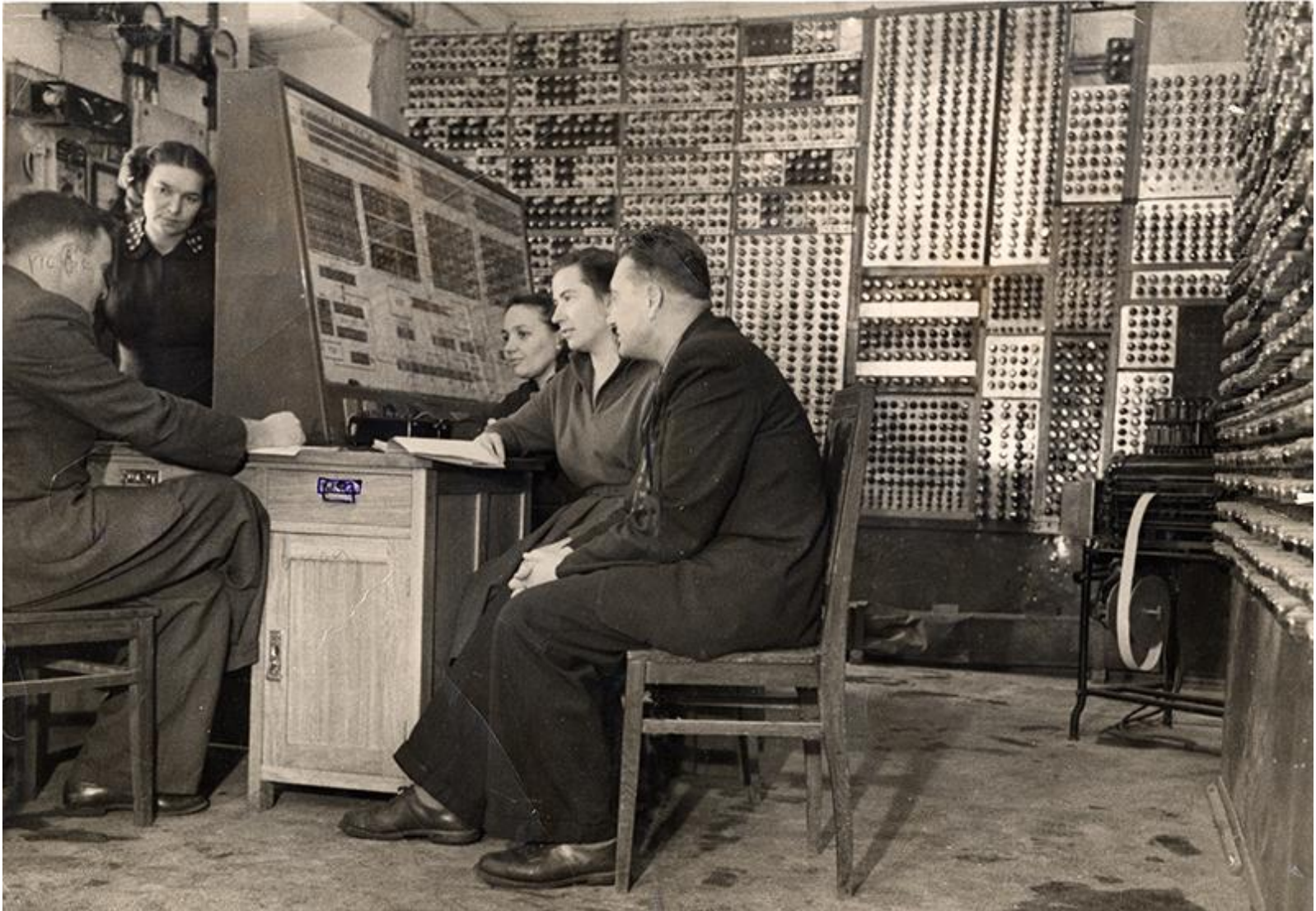
Електронна вакуумна лампа



Glass envelope — скляна колба,
Filament (cathode) — нитка (катод), Grid —
сітка, Plate (anode) — анодна пластина



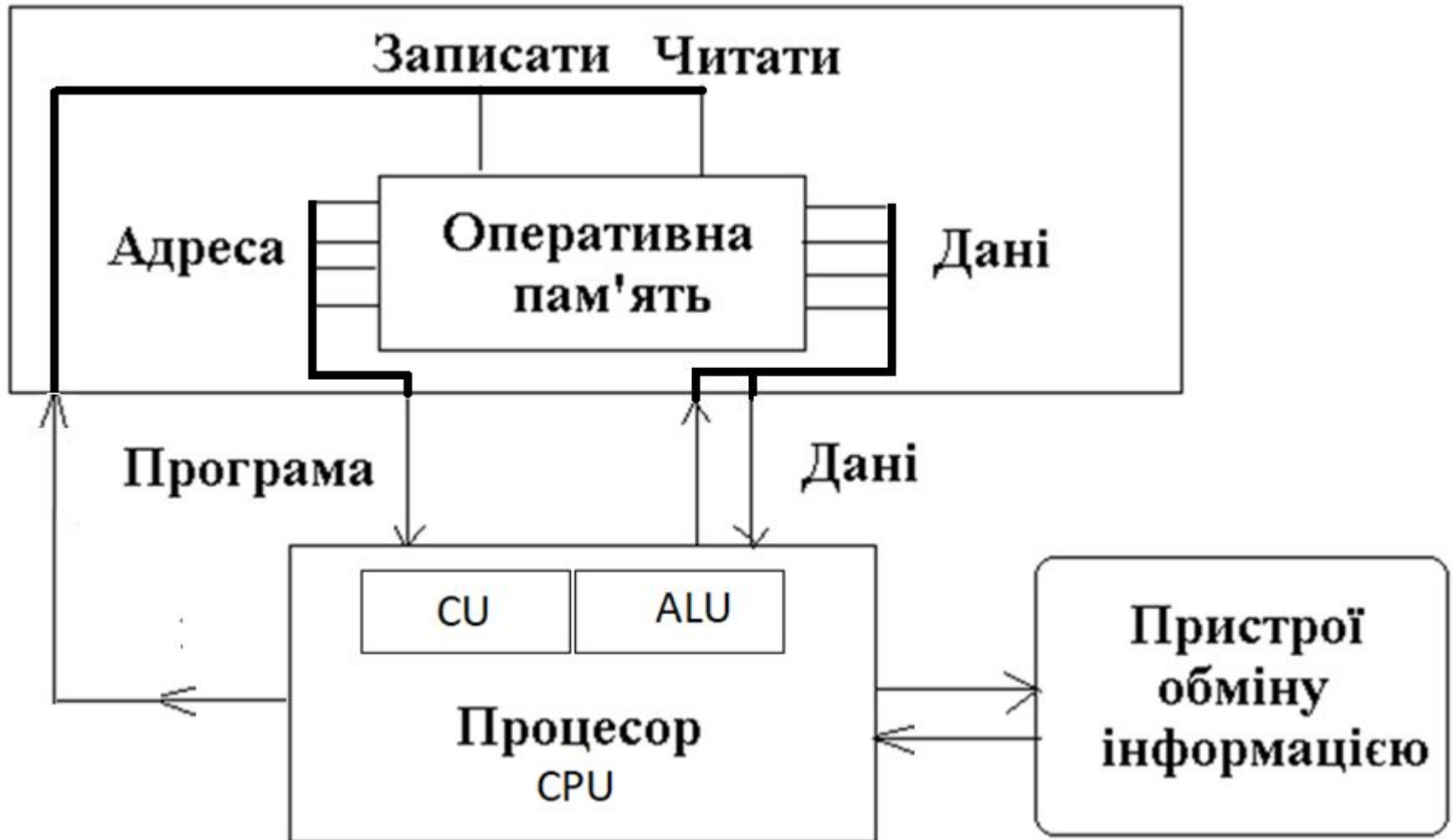
MECM



ЕОМ Урал-2



Структура комп'ютера



Програма для комп'ютера

- Програма для комп'ютера - це послідовність команд для процесора.
- Кожна команда має свою унікальну адресу, що зветься номером команди
- Команда зазвичай складається з коду операції, яку має виконати процесор, та адреси даних, що використовуються у операції. Наприклад, 01 5A2B
«Прочитати дані з комірки пам'яті 5A2B»

Програма для комп'ютера

| Номер команди | Код операції | Адреси даних (операнди) | | |
|---------------|--------------|-------------------------|-----|-----|
| | | 1 | 2 | 3 |
| 0 | Читати | 100 | | |
| 1 | Додати | 200 | | |
| 2 | Записати | 300 | | |
| | | | | |
| 4 | ЧтДдЗп | 100 | 200 | 300 |

Поняття про GoTo

| Номер команди | Код операції | Адреси даних (операнди) | | |
|---------------|--------------|-------------------------|-----|-----|
| | | 1 | 2 | 3 |
| 10 | Якщо <0 | 13 | | |
| 11 | ЧтДдЗп | 100 | 200 | 300 |
| 12 | Перейти | 14 | | |
| 13 | ЧтВдЗп | 100 | 200 | 300 |
| 14 | | | | |

Subroutine and Stack

| Номер команди | Код операції | Адреси даних (операнди) | | |
|---------------|--------------|-------------------------|----|---|
| | | 1 | 2 | 3 |
| 159 | | | | |
| 160 | Call | 500 | | |
| ... | | | | |
| 193 | Call | 500 | | |
| ... | | | | |
| 500 | Щось ... | зроби | ти | |
| ... | | | | |
| 578 | Return | | | |

Питання

- Фон-Неймановська архітектура
- Програма для комп'ютера
- Лічильник команд
- GoTo. Що це, навіщо, проблеми?
- Підпрограма (subroutine)
- Стек. Що це, навіщо, як працює?
- Stack Overflow?

Друге покоління ЕОМ

Особливості другого покоління ЕОМ:

- алгоритмічні мови,
- програмне забезпечення,
- власні контролери у деяких частин,
- транзисторна елементна база



Перші алгоритмічні мови

```
real :: a, b, c, d
```

```
read a, b
```

```
d = a - b
```

```
if (d) M1, M1, M2
```

```
M1: c = sqrt(d)
```

```
goto M3
```

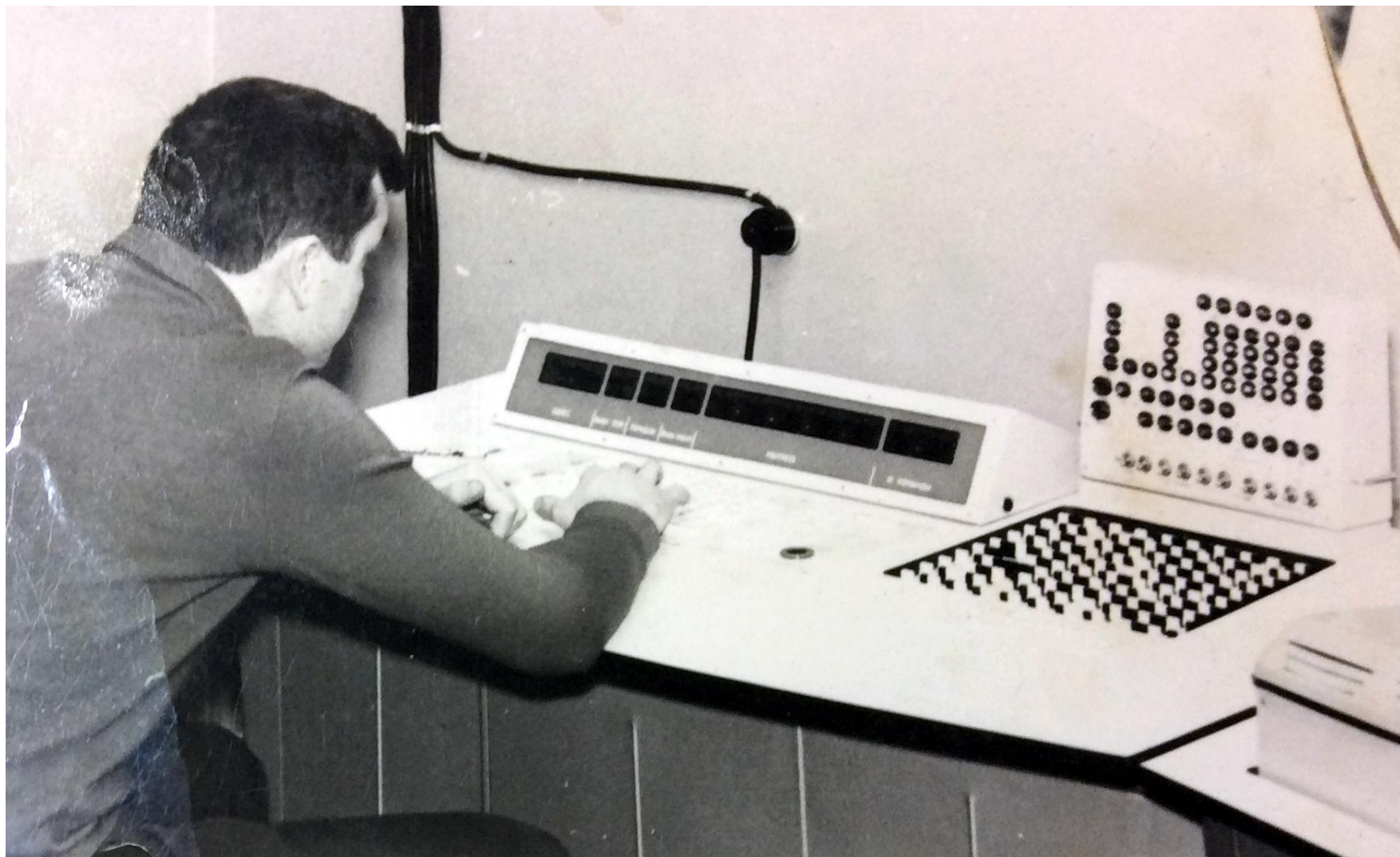
```
M2: c = d * d
```

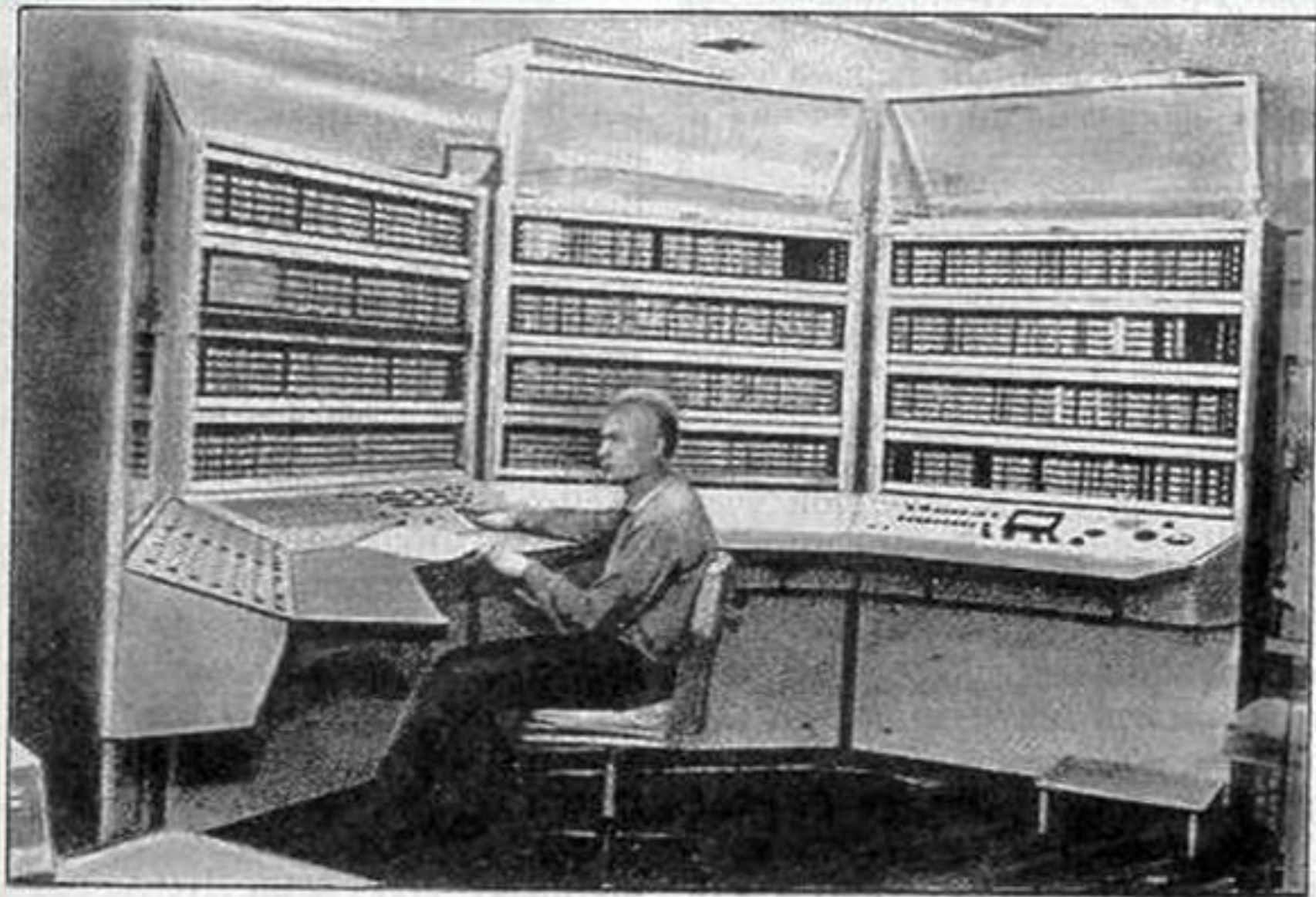
```
M3: write c
```

Переваги мов програмування

- Використання мітки
замість номера команди
- Використання поняття змінна
замість адреси
- Можливість оголошення типу даних
- Використання арифметичних виразів
- Можливість використання бібліотек
стандартних функцій

Міні-ЕОМ Промінь





Компьютер БЭСМ-6

ЕОМ Минск-32



Міні-ЕОМ МИР-2



Питання

- Основні ознаки алгоритмічних мов?
- Недоліки перших алгоритмічних мов?

Третє покоління ЕОМ



Використання інтегральних схем дозволило зробити ЕОМ порівняно компактними, більш потужними, і знизити витрати на виробництво.

Суттєво покращилося програмне забезпечення.

ІВМ 360/370 >>> ЕС ЭВМ

Нововведення IBM

- Байт. Скільки у ньому біт?
- Перфолента на 8 доріжок
- Шістнадцяткова система числення
- 0123 – це скільки?
- 0x123 – це скільки?

Дивні числа?

$$123 = 1 * 100 + 2 * 10 + 3 = 123$$

$$0123 = 1 * 64 + 2 * 8 + 3 = 83$$

$$0x123 = 1 * 256 + 2 * 16 + 3 = 291$$

$$0xABC = 10 * 256 + 11 * 16 + 12 =$$

$$0b1111 = 1*8 + 1*4 + 1*2 + 1 = 0xF = 15$$

EOM EC-1020



Четверте покоління ЕОМ

Сучасні комп'ютери є представниками ЕОМ четвертого покоління.

Елементною базою цих комп'ютерів є

ВІС



Міні-ЕОМ ДВК-2



Міні-ЕОМ Іскра-1030



Прогрес у програмуванні

- DOS >> Windows
- Norton Commander >> Windows
Commander >> *Windows Explorer*
- Електронні таблиці
- Текстові процесори (Word)
- СУБД (DBase, FoxPro)
- ООП (Smalltalk)
- Системи програмування (Delphi, ...)

Системи програмування

Системи програмування – це інтегровані середовища розробки програм, до складу яких входять редактори текстів, дебагери, транслятори та інше

- Delphi
- Visual Studio
- Net Beans, Eclipse, IntelliJ IDEA
- QT-creator

П'яте покоління?



Прогрес у програмуванні

- Інтернет - програмування
- Скриптові мови програмування

Шосте
покоління?



Навіщо
я вам
це
розповідав?

Чи ви готові до цього?

Може це ваше майбутнє?



Історія мови С

Автори: Кен Томпсон

Денніс Рітчі

Створено у 1969—1973

Опубліковано у 1978

Спадкоємці

С++ - 1983

С# - 2001

Цитати

Brian W. Kernighan

(автор першої книги з Сі):

«Сі — інструмент, гострий, як бритва: за його допомогою можна створити і елегантну програму, і кроваве місиво».

Цитати

Просто програміст в Інтернеті

«Мне доводилось писать и на сях++ и на джаве.

Начинал с сей.

Программируя на сях легче облажаться
и не заметить этого.

Человеческий фактор! ...

Поэтому программирование на спп повышает
самооценку

Как же, как же!

Я пишу на спп! Я крутой программер!»

Мову С найчастіше використовують
системні програмісти,
хоча нею можна писати
і прикладні програми

Системне програмування

Створення програм,
які забезпечують керування такими
компонентами комп'ютерної системи як
процесор, оперативна пам'ять, пристрої
введення-виведення, мережеве
обладнання.

Такі програми створюють «інтерфейс»,
с одного боку якого апаратура,
а з іншого — застосування користувача.

Системне програмне забезпечення

Системне програмне забезпечення
на відміну від прикладного,
не вирішує конкретні практичні задачі,
а лише забезпечує роботу інших програм,
і керує апаратними ресурсами
обчислювальної системи.

Програма на мові С

- Програма на мові С – це файл, який має розширення .c, або .cpp, якщо програма написана на С++.
- Файл програми являє собою сукупність різноманітних оголошень та функцій, які розташовуються певним чином.

Програма на мові С

```
#include <QtCore/QCoreApplication>
#include <iostream>
#include <cmath>
using namespace std;
int main(int argc, char *argv[])
{
    QCoreApplication a(argc, argv);
    cout << "Привіт першокурснику!\n";
    cout << "Qt вміє рахувати:\n"
         << "2+3 = " << 2+3 << "\n";
    cout << "Qt знає тригонометрію:\n"
         << "sin(0.5) = " << sin(0.5) << "\n";
    cout << "Qt вміє добувати корені:\n"
         << "корінь квадратний із 2 дорівнює "
         << sqrt(2) << "\n";
    return a.exec();
}
```

Оголошення

Ф
У
Н
К
Ц
І
Я

```
#include <QtCore/QCoreApplication>
#include <iostream>
#include <cmath>
using namespace std;
int main(int argc, char *argv[])
{
    QCoreApplication a(argc, argv);
    cout << "Привіт першокурснику!\n";
    cout << "Qt вміє рахувати:\n"
         << "2+3 = " << 2+3 << "\n";
    cout << "Qt знає тригонометрію:\n"
         << "sin(0.5) = " << sin(0.5) << "\n";
    cout << "Qt вміє добувати корені:\n"
         << "корінь квадратний із 2 дорівнює "
         << sqrt(2) << "\n";
    return a.exec();
}
```

Оголошення

Ф
У
Н
К
Ц
І
Я

Функція `main()`

Кожна програма на C та C++
обов'язково

повинна мати у своєму складі
функцію з назвою **`main()`**.

Це головна функція програми з якої
починається виконання кожної програми.

Функція main()

Тип значення,
що повертає
функція

Заголовок
функції

Список формальних
параметрів

```
int main(int argc, char *argv[])  
{
```

Тіло
функції

```
    QApplication a(argc, argv);  
    cout << "Привіт першокурснику!\n";  
    return a.exec();  
}
```


Функція main()

У найпростішому випадку функція main() може мати і такий вигляд:

```
int main(void) {  
    ...  
    return 0;  
}
```

Об'єкт cout

cout – це об'єкт C++,
що відповідає за виведення інформації
на консоль.

Операція << означає,
що значення змінної,
або результат обчислення виразу
буде перетворено у символи
та передано об'єкту cout.

Об'єкт cout

```
cout << "Привіт першокурснику!\n";  
cout << "Qt вміє рахувати:\n"  
    << "2+3 = " << 2+3 << "\n";  
cout << "Qt знає тригонометрію:\n"  
    << "sin(0.5) = " << sin(0.5) << "\n";
```

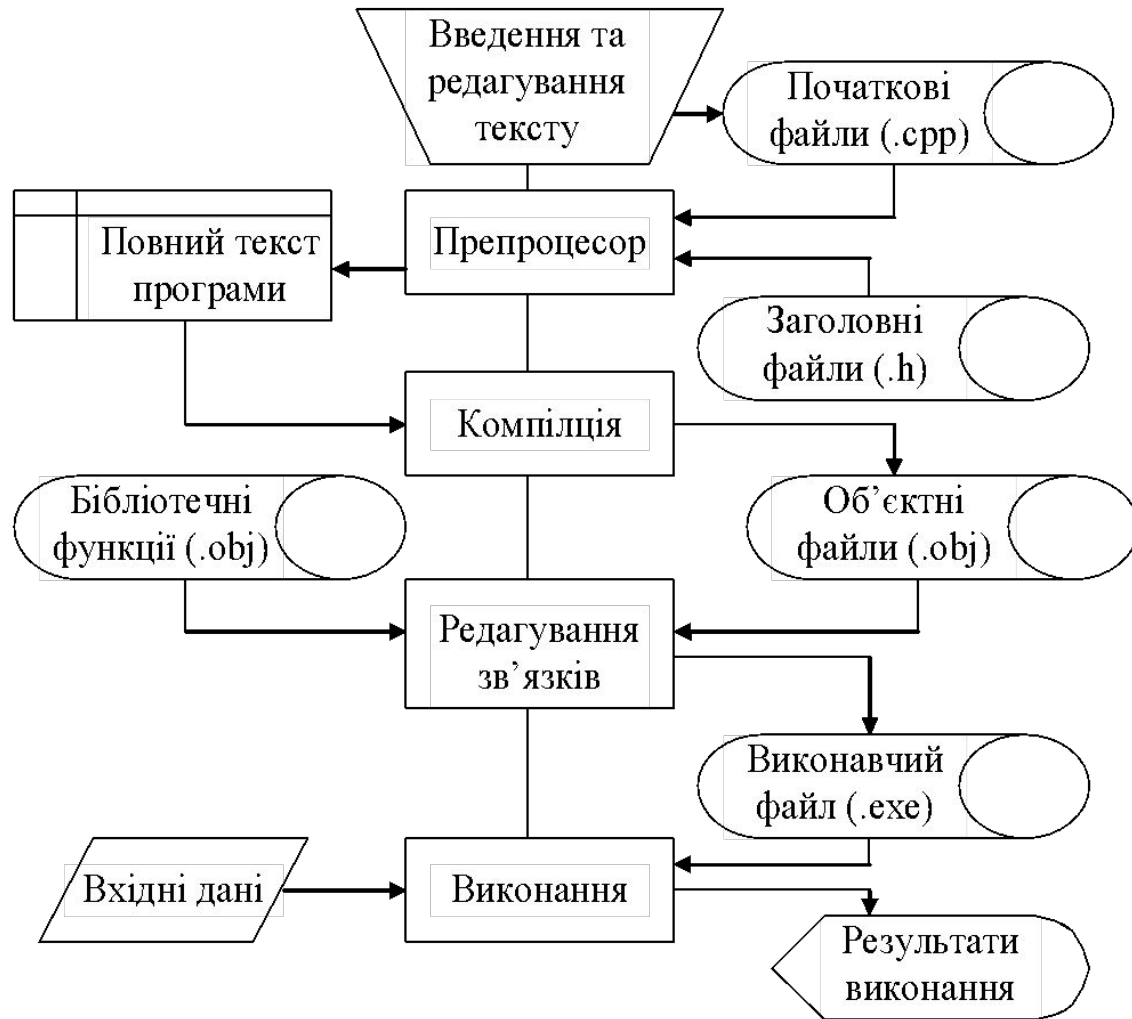
Об'єкт `cin`

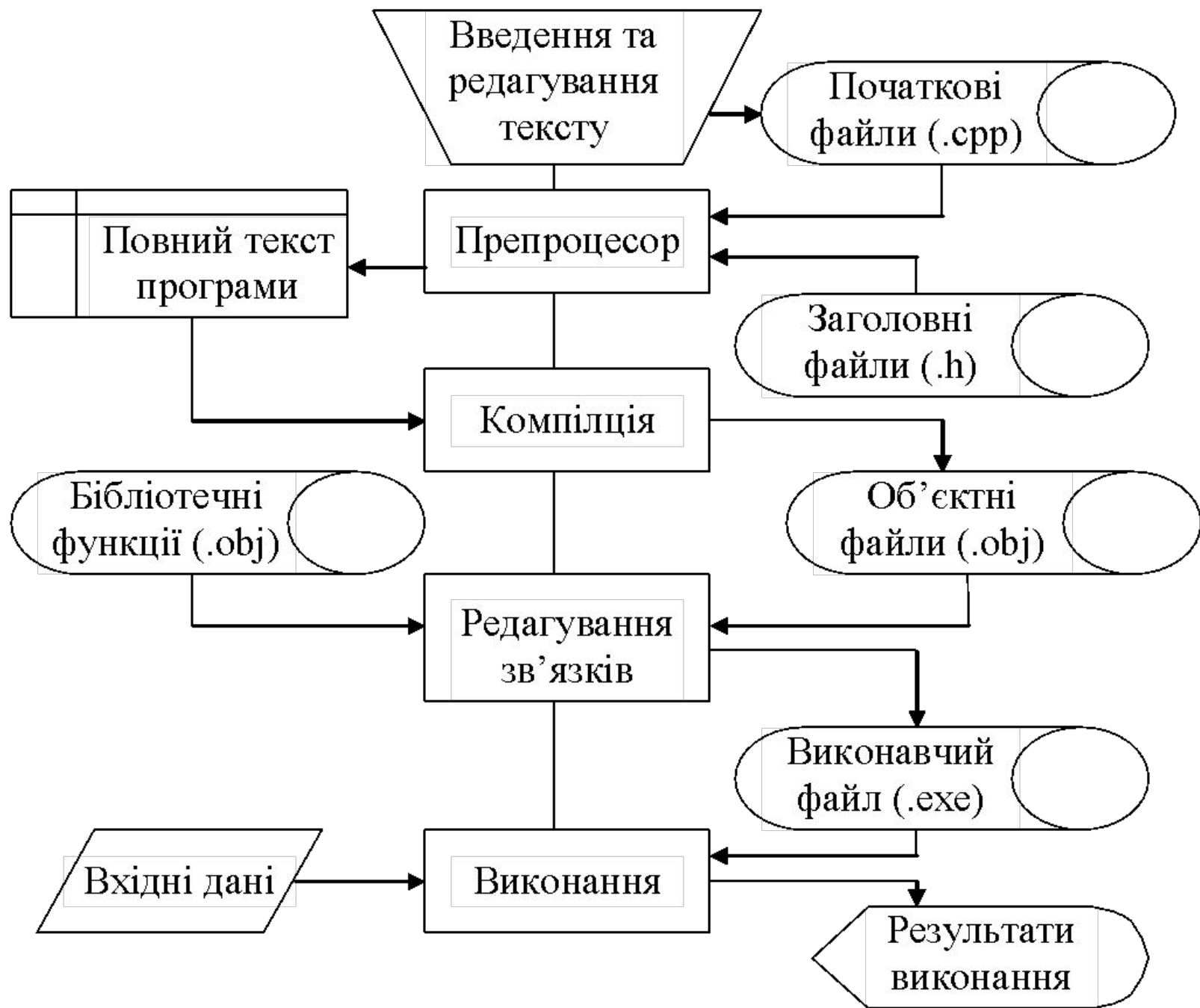
Об'єкт `cin` представляє собою потік символів з клавіатури у програму. Операція `>>` бере інформацію з цього потоку і перетворює її відповідно до типу змінної, що отримує цю інформацію.

Об'єкт cin

```
string name;  
int a, b; double c;  
cin >> name;  
cin>>a>>b  
>>c;
```

Етапи виконання програми





Елементи мови Сі

- символи (алфавіт мови) - це основні неподільні знаки, за допомогою яких пишуться всі тексти на мові програмування;
- лексеми (слова) - мінімальні одиниці мови, які мають самостійний зміст;
- вирази - задають правило обчислення деякого значення;
- оператори (інструкції) задають опис деякої дії.

Алфавіт мови

- великі та малі латинські літери: A-Z, a-z.
- арабські цифри;
- символи: графічні та ескейп-послідовності (символи табуляції, символ переходу на наступний рядок тощо);
- СИМВОЛИ , . ; : ? ' ! | / \ ~ () [] { } < > # % ^ & - + * =

Лексеми

Ключові слова

| | | | |
|-----------------------|---------------------|-----------------------|-----------------------|
| <code>auto</code> | <code>double</code> | <code>int</code> | <code>struct</code> |
| <code>break</code> | <code>else</code> | <code>long</code> | <code>switch</code> |
| <code>case</code> | <code>enum</code> | <code>register</code> | <code>typedef</code> |
| <code>char</code> | <code>extern</code> | <code>return</code> | <code>union</code> |
| <code>const</code> | <code>float</code> | <code>short</code> | <code>unsigned</code> |
| <code>continue</code> | <code>for</code> | <code>signed</code> | <code>void</code> |
| <code>default</code> | <code>goto</code> | <code>sizeof</code> | <code>volatile</code> |
| <code>do</code> | <code>if</code> | <code>static</code> | <code>while</code> |

Лексеми

Ідентифікатори

Ідентифікатори – це імена змінних, функцій та деяких інших об'єктів програми.

- Мають починатися з букви
- Можуть містити цифри
- Мають писатися без пропусків

Лексеми

Ідентифікатори

У Сі великі і маленькі літери,
що використовуються для імен,
розрізняються!

але,

у С дуже любляють знак підкреслення.
max_int_number замість **maxIntNumber**

Лексеми

Константи цілочислові

- Десяткові: 12345 -17 +1098
- Вісімкові: 0110 077 01234567
- Шістнадцяткові: 0x123 0xF3a2E

Лексеми

Константи цілочислові

- Десяткові:

$$1234 = 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0$$

- Вісімкові:

$$01234 = 1 \cdot 8^3 + 2 \cdot 8^2 + 3 \cdot 8^1 + 4 \cdot 8^0$$

- Шістнадцяткові:

$$0x123 = 1 \cdot 16^2 + 2 \cdot 16^1 + 3 \cdot 16^0$$

Двійкові числа

0 (0), 1 (1), 10 (2), 11 (3), 100 (4), 101 (5),
110 (6), 111 (7), 1000 (8), 1001 (9), 1010 (A),
1011 (B), 1100 (C), 1101 (D), 1110 (E), 1111 (F)

0x3b2c=0011 11011 0010 1100;

101110=

$$1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 =$$
$$32 + 8 + 4 + 2 = 46$$

10 -> 2

| | Частка | Залишок |
|--------|--------|---------|
| 46 : 2 | 23 | 0 |
| 23 : 2 | 11 | 1 |
| 11 : 2 | 5 | 1 |
| 5 : 2 | 2 | 1 |
| 2 : 2 | 1 | 0 |
| 1 : 2 | 0 | 1 |

10 -> 16 -> 2

| | Частка | Залишок | 16 | 2 |
|---------|--------|---------|----|--------|
| 46 : 16 | 2 | 14 | Е | 1110 ↑ |
| 2 : 16 | 0 | 2 | 2 | 10 ↑ |

Лексеми

Дійсні константи

- З фіксованою крапкою:
1.2345 -0.017 +1098.2345
- З плаваючою крапкою
123.45e-2 1.0982345E3

Лексеми

Символьні константи

- Прості символи:
'A' 'b' 'щ' '[' '\'
- Ескейп-послідовності
'\n' '\a' '\"' '\t'

Лексеми

Символьні рядки

"Привіт першокурснику!\n"

"Ми вивчаємо курс \"Програмування на C\"!"

Лексеми

Знаки операцій, роздільники, коментарі

/* Односимвольні операції:*/

+ * < ^ / ...

// Багатосимвольні операції:

++ || >> += <= <<=

// Роздільники:

() [] { }, ; : = * #