

Базовые управляющие конструкции

Лекция 3-2

Линейный вычислительный процесс

- Инструкция — выполняемая единица, имеющая смысл и ограниченная символом «;»
- Последовательное выполнение инструкций друг за другом

```
int x;  
x = 15;  
x = input();  
if (x == 15) x = 0;  
print(x);
```



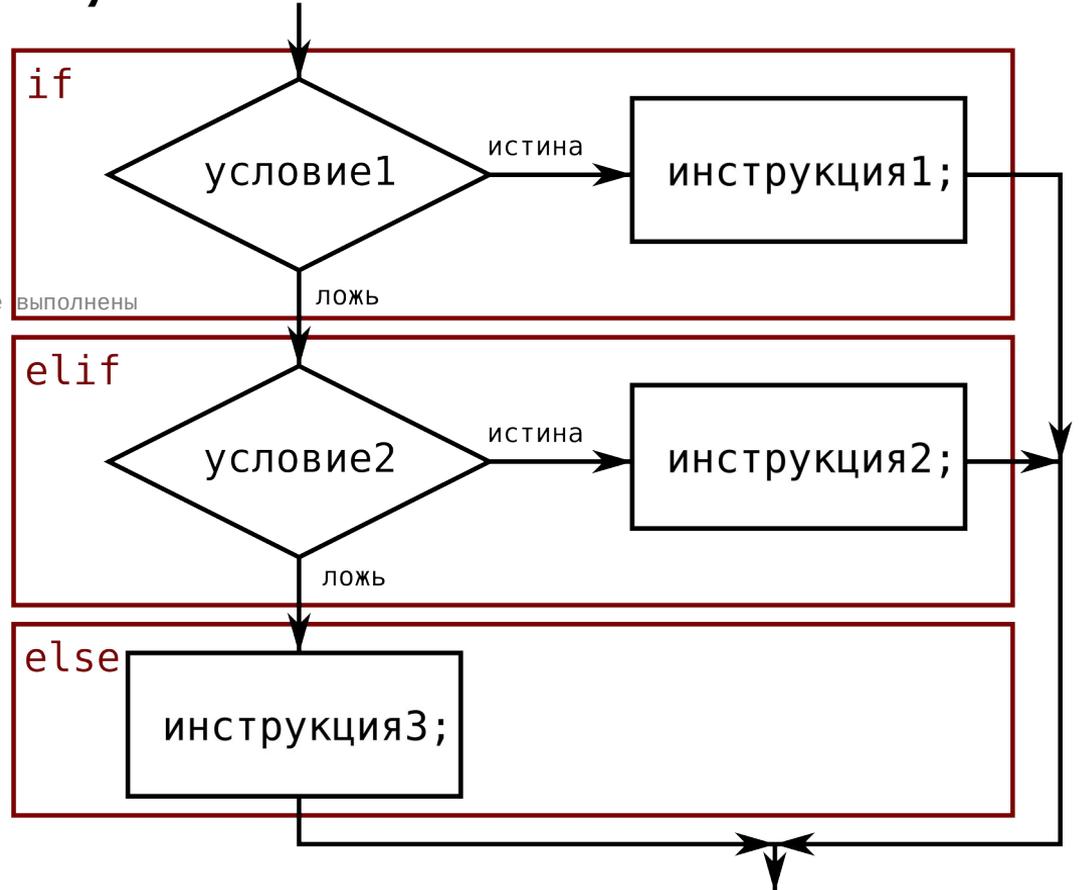
Разветвленный вычислительный процесс

Выполнение или невыполнение некоторого набора команд должно зависеть от выполнения или невыполнения некоторого условия

- Условный оператор `if`
- Оператор многозначного выбора `switch`

if, elif, else

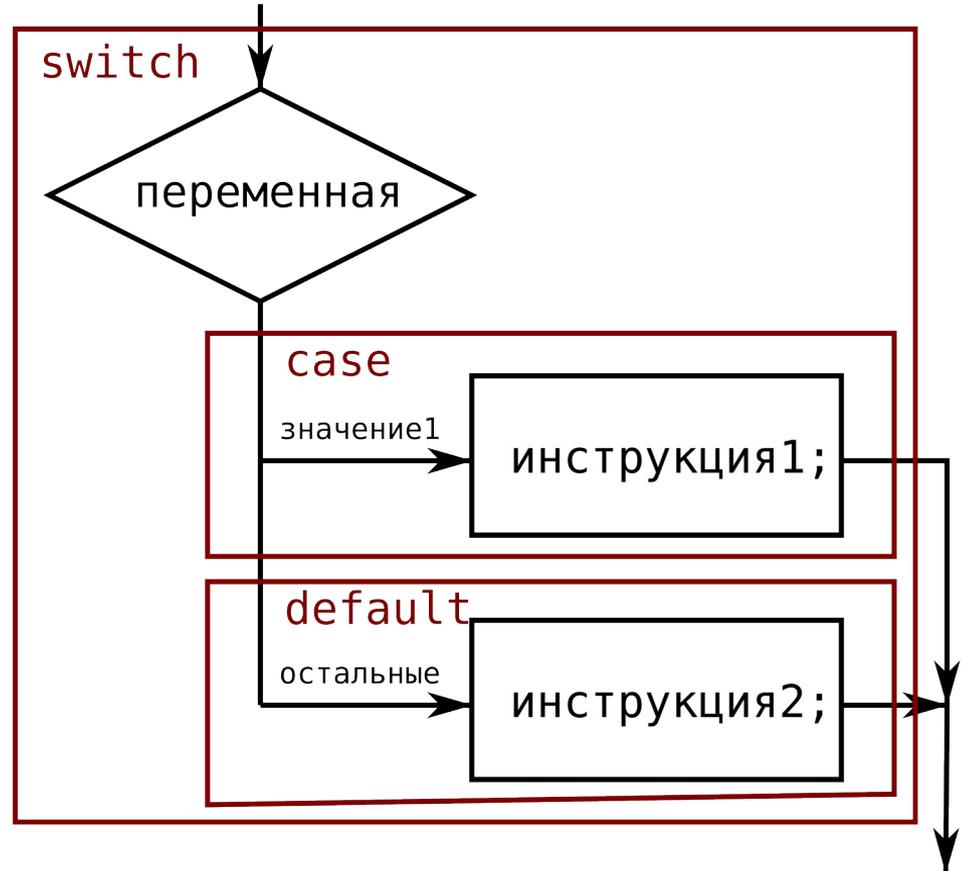
```
if (условие1) { // обязательный элемент
    блок1
}
elif (условие2) { // если все предыдущие условия не выполнены
    блок2
}
else { // если ни одно условие не выполнено
    блок3
}
```



switch

```
switch (переменная) {  
    case 1: блок1; break;  
    case 2: блок2; break;  
    case 3: блок3; break;  
    default: блок4; break;  
}
```

Switch поочередно сравнивает значение переменной со значениями в каждом блоке case. Если совпадение обнаружено, то выполняется соответствующий блок case. Если совпадений не обнаружено, то выполняется блок default.



Циклический вычислительный процесс

Цикл — разновидность управляющей конструкции, предназначенная для организации многократного исполнения набора инструкций

- Цикл со счетчиком `for`
- Цикл с предусловием `while`
- Цикл с постусловием `do...while`

for

```
for (инициализация; условие; приращение) {  
    тело цикла;  
}
```

```
for (int i = 0; i < 10; i++) {  
    print(i);  
}
```

Цикл for создает переменную.
В каждой итерации проверяется условие.
Если условие истинно, то тело цикла выполняется.
Затем выполняется приращение.
После этого снова проверяется условие.
При неправильной организации цикл может превратиться в бесконечный



while

```
while (условие) {  
    тело цикла;  
}
```

```
int x = 1;  
while (x < 100) {  
    x *= 2 * input();  
}
```

В каждой итерации проверяется условие.

Если условие истинно, то тело цикла выполняется.

После этого снова проверяется условие.

При неправильной организации цикл может превратиться в бесконечный.

Используется когда приращение заранее непредсказуемо или зависит от

других факторов.



do...while

```
do {  
    тело цикла;  
} while (условие);
```

```
int x = 100;  
do {  
    x *= 2 * input();  
} while (x < 100);
```

Выполняется тело цикла.

Если условие истинно, то тело цикла выполняется снова.

При неправильной организации цикл может превратиться в бесконечный.

Используется когда приращение заранее непредсказуемо или зависит от других факторов, при этом тело должно быть выполнено хотя бы один раз.

