Семинар 2 по УП

Условен оператор. Цикли.

Условен оператор

- Ключова дума if
- Ако …, то направи …
- В противен случай (ключова дума else)

Малко код (Наистина малко)

```
if(/*условие*/)
{
    //Ако условието е изпълнено направи това
}
else
{
    //В противен случай направи това.
}
```

Какво забелязахме

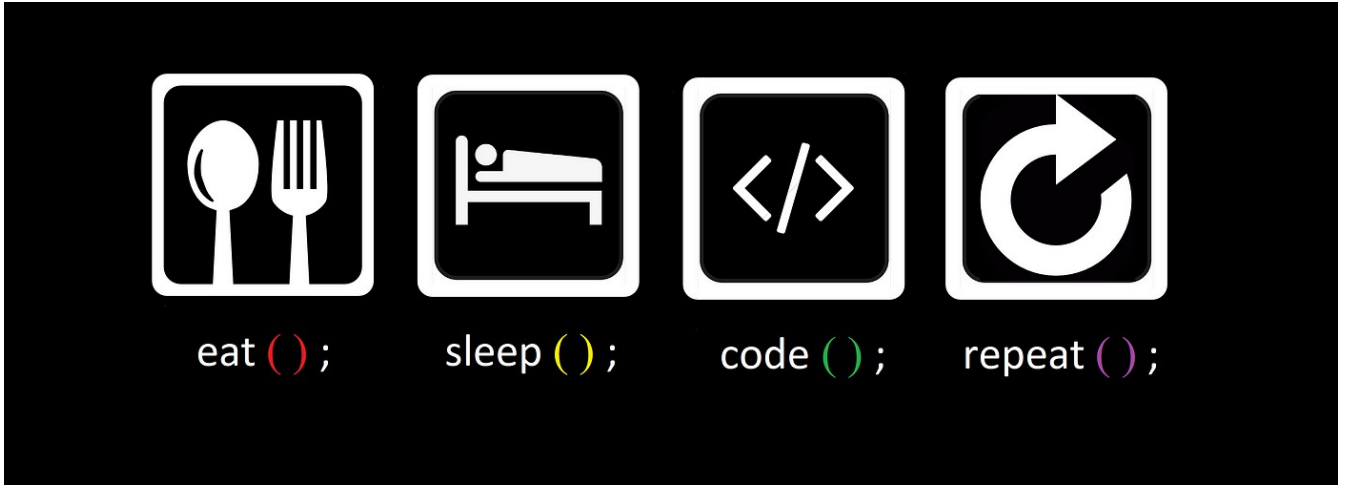
- На мястото на */*условие*/ седи валиден булев израз. Той може да е булева променлива. Може дори да е константа ако ви се правят тавтологии.*
- *В блоковете седят фрагменти код, които да се изпълнят при съответния сценарии.*

“Много трудни” въпроси

- Какво ще е условието за да проверим дали едно число е четно?
- А дали число завършва на 1?
- Дали сумата на 2 числа е равна на 3то?
- При сравняване на числа какво използваме?

Цикъл

- Защото програмистите сме мързеливи хора.
- Защото може да не знаем колко пъти трябва да извършим дадено действие.



Цикъл While

- Повтаря докато нещо е вярно.
- Първокласен цикъл (защото дори първокласник може да го подкара)
- Обикновено се използва, когато нямаме ясни граници от къде до къде ще въртим този цикъл.
- Просто въртим и чакаме нещо да спре да е вярно.

Още малко код

```
while(/*условие*/)
{
    //Какво да пръвим ако това условие е вярно
}
```

- Мемето от преди малко, само че с while цикъл. Относително смислен пример, защото никой не знае докога ще е жив.

```
while (alive) {
  eat();
  sleep();
  code();
  repeat();
}
```

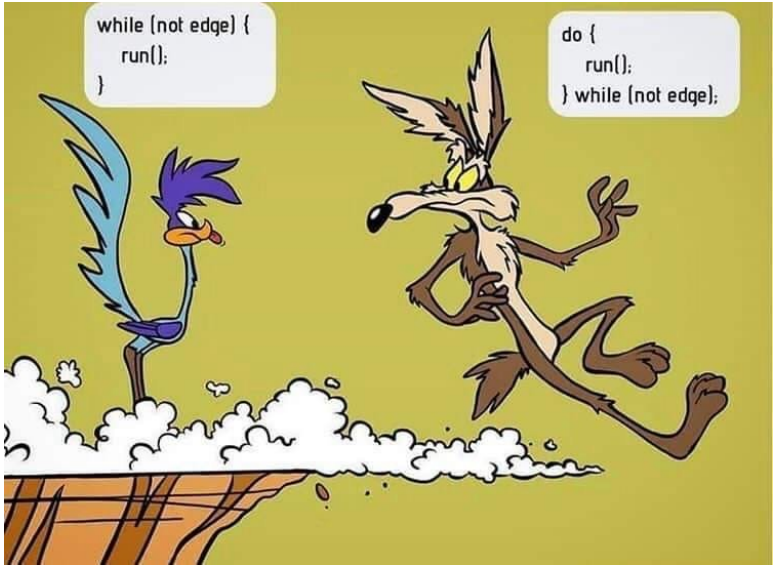
Сега учебникарски пример

- Валидация на входните данни или когато потребителят не ни помага, а ни пречи (винаги).

```
int num = -1;
while(num < 0)
{
    std::cout << "Enter a positive number: ";
    std::cin >> num;
}
```

Цикъл Do While

- Подобно на while, само че при do while първо се изпълнява блока, а после се проверява условието и така си гарантираме, че поне един път ще се изпълни кодът в блока.



Същия учебникарски пример

```
int num;
do
{
    std::cout << "Enter a positive number: ";
    std::cin >> num;
}
while(num < 0);
```

- Преди трябваше num да се инициализира с невалидна стойност, за да се гарантира поне едно изпълнение на блока, докато тук не е така
- Сложихме ; след while(/*условие*/)

Цикъл For

- Използваме този цикъл, когато условието зависи по някакъв начин от някаква управляваща променлива.
- Тази променлива се инициализира в самата дефиниция на цикъла и стъпката и на промяна също се описва в цикъла.

Код

```
for(/*инициализация*/; /*условие*/; /*стъпка*/)
{
    //Нещо което да правим
}
```

- Почти равносилно е на:

```
//инициализация
while(/*условие*/)
{
    //Нещо което да правим
    //стъпка
}
```

Пример

- Извеждане на числата от a до b като a и b са ни въведени от потребителя.

```
int a, b;
std::cin >> a >> b;
for(int i = a; i <= b; i++)
{
    std::cout << i << " ";
}
std::cout << std::endl;
```

Задачи

- 1. Изведете сумата на числата от a до b през 2 стъпки, а през c стъпки?
- 2. Да се напише програма, въвежда от клавиатурата естестествено число X и проверява дали има естествено число Y, за което Y! == X. Например, за числото 24 (=4!), отговорът е “да”, а за числото 20 – “не”. (2 задача от Контролно 1 вариант 1 миналата година)
- 3. Въвеждаме цяло положително число. Да се изведе true, ако то е съставено само от четни цифри и false, иначе. (Почти задача 1 от Контролно 2 вариант 2 миналата година. Леко я бутнах да става за вашите знания)
- 4. Изведете всички четни числа в интервал [a,b].
- 5. Въвеждат се n години. Пребройте колко от въведените години са високосни и изведете броя им на екрана.

Благодаря ви за вниманието!