

Референции

- Синтаксис:

<тип> & <име> = <име на друга променлива от същия тип>

Предназначение:

- Променлива, която притежава данните на вече съществуваща променлива

Пояснение

- Все едно дадената променлива вече има две имена

Пример

```
int Mama = 50;           //Мама дава 50 лева за коледа
int Tati = Mama;        //Тати дава колкото мама
Mama -= 30;
std::cout<<Mama<<', '<<Tati; //20,50
```

```
int Mama = 50;           //Мама дава 50 лева за коледа
int & Tati = Mama;       //оказва се, че това е общ подарък 😞
Mama -= 30;
std::cout<<Mama<<', '<<Tati; //20,20
```

Важно

- Също както константите, референцията трябва да се инициализира още при дефиницията
- След като веднъж е била декларирана, инициализирането е необратимо (обвързване за цял живот)
- Типът на референцията и на променливата трябва да съвпадат
- При инициализация не се копират данните на оригиналната променлива ([Demo1](#)) => пести се време и памет при огромни структури от данни

Обратно в началото

Как да накараме функцията да работи?

```
void swap(double a, double b)
```

```
{
```

```
    double c = a;
```

```
    a = b;
```

```
    b = c;
```

```
}
```

Обратно в началото

Как да накараме функцията да работи?

```
void swap(double & a, double & b)
```

```
{
```

```
    double c = a;
```

```
    a = b;
```

```
    b = c;
```

```
}
```

- Demo2

Задача от миналия път

- Какво ще изведе следната програма?

```
void Abs(int a)
{
    if(a<0)
        a*=-1;
}
int numb = -5;
Abs(numb);
std::cout<<numb;
```

- Отговор: -5

Задача от миналия път Remastered

- Какво ще изведе следната програма?

```
void Abs(int & a)
{
    if(a<0)
        a*=-1;
}
int numb = -5;
Abs(numb);
std::cout<<numb;
```

- Отговор: 5, вече работи така както искаме

Пример

- Нека имаме следната функция

```
void example(int & numb)
{
    numb+=5;
}
```


Пример

```
int main()
{
    int a = 5;
    example(a);
    return 0;
}
```


