

Примеры оптимизаций

Подстановка (Inlining)

До оптимизации

```
double Sqr(double x)
{
    return x*x;
}
static void Main()
{
    ...
    double d = Sqr(y);
}
```

После оптимизации

```
static void Main()
{
    ...
    double d = y * y;
}
```

Свёртка констант (Constant Folding)

До оптимизации

```
x = 2*2+1;
```

После оптимизации

```
x = 5;
```

Применение алгебраических тождеств

До оптимизации

```
x = y + 0;
k = m * 1;
```

После оптимизации

```
x = y;
k = m;
```

Распространение копий (Copy Propagation)

До оптимизации

```
int calc(int x, int y)
{
    int a = x;
    int b = y;
    return a * a + b * b;
}
```

После оптимизации

```
int calc(int x, int y)
{
    return x * x + y * y;
}
```

Распространение констант (Constant Propagation)

До оптимизации

```
xz = 777;
a = b;
c = a;
d = c + xz;
```

После оптимизации

```
xz = 777;
a = b;
c = a;
d = c + 777;
```

Удаление мертвого кода (Dead Code Elimination)

До оптимизации

```
{
    ...
    foo();
    xz = 911;
    for (int i=0;...)
        a[i] = b[i];
}
void foo()
{
    while (true){};
}
```

После оптимизации

```
{
    ...
    foo();
    // ничего
}
void foo()
{
    while (true){};
}
```

Удаление неиспользуемых присваиваний

До оптимизации

```
a = 5;
a = 6;
b = a + 1;
```

После оптимизации

```
a = 6;
b = a + 1;
```

Удаление общих подвыражений (CSE – Common Subexpressions Elimination)

До оптимизации

```
a = x + y;
b = x + y;
```

После оптимизации

```
t = x + y;
a = t;
b = t;
```

Взрыв объектов (Object Explosion)

До оптимизации

```
class Vec
{
    int x;
    int y;
    Vec (int d)
    {
        x = y = d;
    }
}
int foo(int a)
{
    Vec v = new Vec (a);
    return v.x + 2;
}
```

После оптимизации

```
class Vec
{
    int x;
    int y;
    Vec (int d)
    {
        x = y = d;
    }
}
int foo(int a)
{
    int x = a;
    int y = a;
    return x + 2;
}
```

Вынос инвариантного кода (Invariant Code Motion)

До оптимизации

```
for(int k=0; k<N; k++)
    a[k] = b[i];
```

После оптимизации

```
t = b[i];
for(int k=0; k<N; k++)
    a[k] = t;
```

Удаление частичной избыточности (PRE – Partial Redundancy Elimination)

До оптимизации

```
a = b*3;
if(j < 0)
    print(a);
return;
```

После оптимизации

```
if (j < 0)
{
    a = b*3;
    print(a);
}
return;
```

Выявление индуктивных переменных (Induction Variables)

До оптимизации

```
for (i = 0; i < 10; i++)
{
    j = 17 * i;
}
```

После оптимизации

```
j = -17;
for (i = 0; i < 10; i++)
{
    j = j + 17;
}
```