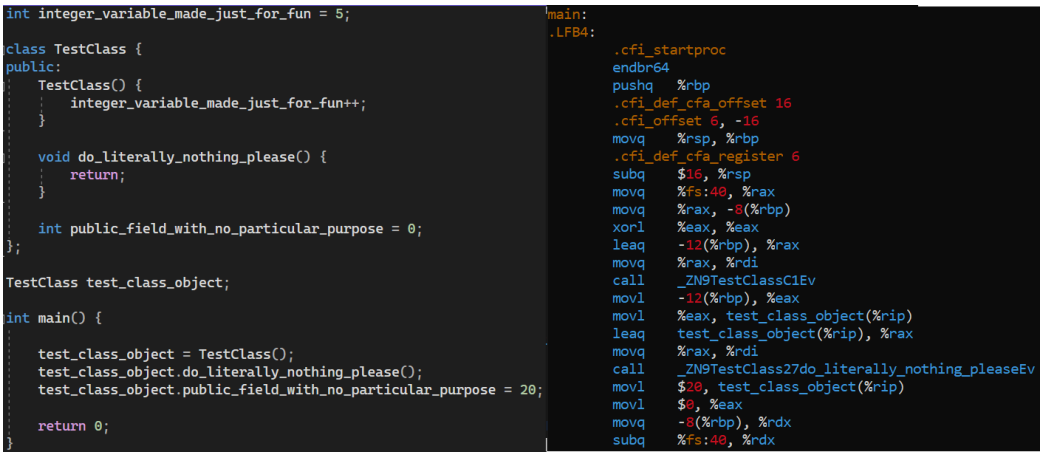
Лабораторная работа 6

Калашников Михаил, Б03-205

Изнанка C++

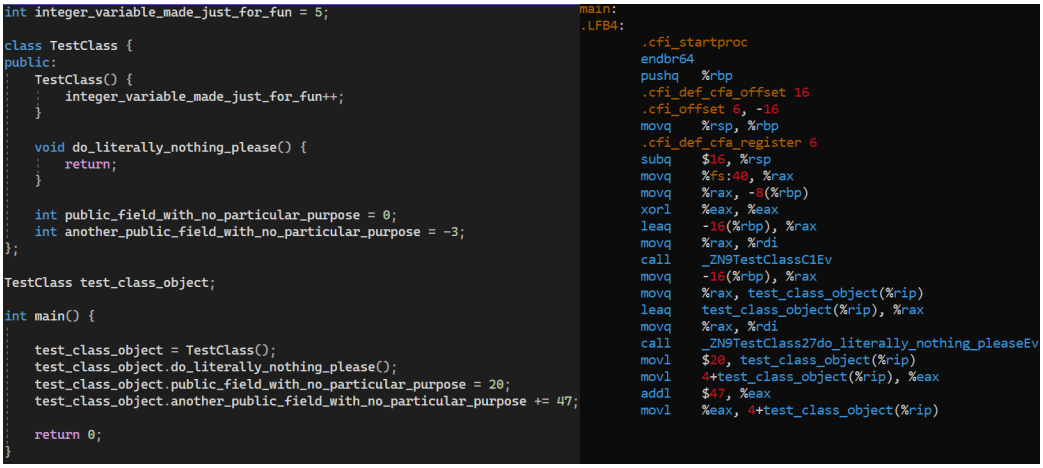
1. Создадим простой класс.



```
int integer_variable_made_just_for_fun = 5;

class TestClass {
public:
    TestClass() {
        integer_variable_made_just_for_fun++;
    }

    void do_literally_nothing_please() {
        return;
    }

    int public_field_with_no_particular_purpose = 0;
};

TestClass test_class_object;

int main() {

    test_class_object = TestClass();
    test_class_object.do_literally_nothing_please();
    test_class_object.public_field_with_no_particular_purpose = 20;

    return 0;
}
```

```
main:
.LFB4:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        subq    $16, %rsp
        movq    %fs:40, %rax
        movq    %rax, -8(%rbp)
        xorl    %eax, %eax
        leaq    -12(%rbp), %rax
        movq    %rax, %rdi
        call    _ZN9TestClassC1Ev
        movl    -12(%rbp), %eax
        movl    %eax, test_class_object(%rip)
        leaq    test_class_object(%rip), %rax
        movq    %rax, %rdi
        call    _ZN9TestClass27do_literally_nothing_pleaseEv
        movl    $20, test_class_object(%rip)
        movl    $0, %eax
        movq    -8(%rbp), %rdx
        subq    %fs:40, %rdx
```

Можно увидеть что название метода просто прилепляется к названию класса. Изменение поля выглядит интересно. Обращение к нему происходит через глобальную переменную экземпляра. Реализации класса и метода выглядит как реализация обычных функций.

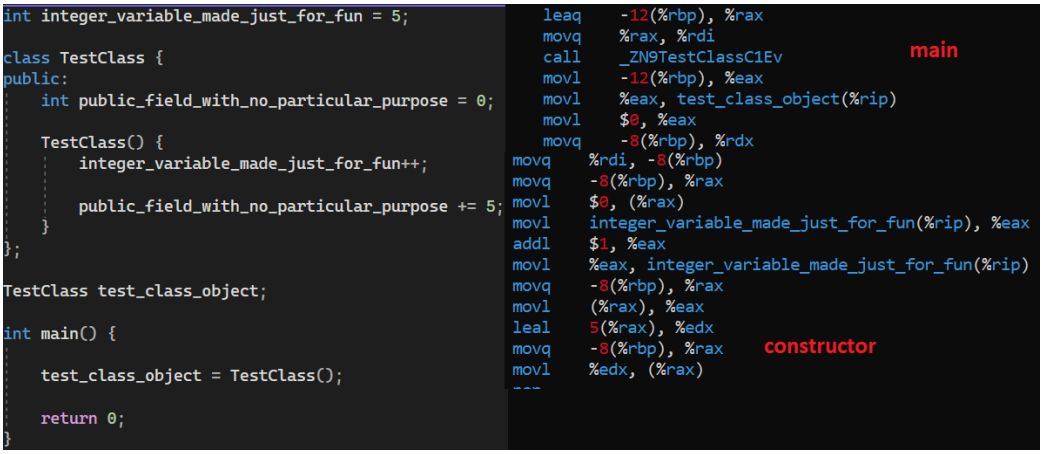


```
int integer_variable_made_just_for_fun = 5;

class TestClass {
public:
    TestClass() {
        integer_variable_made_just_for_fun++;
    }

    void do_literally_nothing_please() {
        return;
    }

    int public_field_with_no_particular_purpose = 0;
    int another_public_field_with_no_particular_purpose = -3;
};

TestClass test_class_object;

int main() {

    test_class_object = TestClass();
    test_class_object.do_literally_nothing_please();
    test_class_object.public_field_with_no_particular_purpose = 20;
    test_class_object.another_public_field_with_no_particular_purpose += 47;

    return 0;
}
```

```
main:
.LFB4:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        subq    $16, %rsp
        movq    %fs:40, %rax
        movq    %rax, -8(%rbp)
        xorl    %eax, %eax
        leaq    -16(%rbp), %rax
        movq    %rax, %rdi
        call    _ZN9TestClassC1Ev
        movq    -16(%rbp), %rax
        movq    %rax, test_class_object(%rip)
        leaq    test_class_object(%rip), %rax
        movq    %rax, %rdi
        call    _ZN9TestClass27do_literally_nothing_pleaseEv
        movl    $20, test_class_object(%rip)
        movl    4+test_class_object(%rip), %eax
        addl    $47, %eax
        movl    %eax, 4+test_class_object(%rip)
```

Добавив второе поле, можно понять, что поля лежат в памяти друг за другом по адресу экземпляра. В принципе, предсказуемо.

2. Указание на конкретный экземпляр класса происходит через регистр %rdi. Перед вызовом методов туда складывается адрес нужного объекта и манипуляции в методе происходят уже с ним.



```
int integer_variable_made_just_for_fun = 5;

class TestClass {
public:
    int public_field_with_no_particular_purpose = 0;

    TestClass() {
        integer_variable_made_just_for_fun++;

        public_field_with_no_particular_purpose += 5;
    }
};

TestClass test_class_object;

int main() {

    test_class_object = TestClass();

    return 0;
}
```

```
    leaq    -12(%rbp), %rax
    movq    %rax, %rdi
    call    _ZN9TestClassC1Ev                    main
    movl    -12(%rbp), %eax
    movl    %eax, test_class_object(%rip)
    movl    $0, %eax
    movq    -8(%rbp), %rdx
movq    %rdi, -8(%rbp)
movq    -8(%rbp), %rax
movl    $0, (%rax)
movl    integer_variable_made_just_for_fun(%rip), %eax
addl    $1, %eax
movl    %eax, integer_variable_made_just_for_fun(%rip)
movq    -8(%rbp), %rax
movl    (%rax), %eax
leal    5(%rax), %edx
movq    -8(%rbp), %rax      constructor
movl    %edx, (%rax)
```

3. Добавление слова this не меняет листинг. По сути, this, т.е. указатель на используемый экземпляр, лежит в регистре %rdi.

4. Что-то вызов дестракторов я не нашел.

5. Приватные и пабликовые методы полностью идентичные. Наличие права обращения к конкретному методу проверяется компилятором, а не в рантайме.

```
_ZN9TestClass13public_methodEv:
.LFB3:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movq    %rdi, -8(%rbp)
        movl    integer_variable_made_just_for_fun(%rip), %eax
        addl    $1, %eax
        movl    %eax, integer_variable_made_just_for_fun(%rip)
        nop
        popq    %rbp
        .cfi_def_cfa 7, 8
        ret
```

```
_ZN9TestClass14private_methodEv:
LFB4:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movq    %rdi, -8(%rbp)
        movl    integer_variable_made_just_for_fun(%rip), %eax
        addl    $1, %eax
        movl    %eax, integer_variable_made_just_for_fun(%rip)
        nop
        popq    %rbp
        .cfi_def_cfa 7, 8
        ret
```

6. При наследовании конструкторы вызываются последовательно в обратном порядке, все как рассказывала Катерина Алексеевна в прошлом семестре.

```
int integer_variable_made_just_for_fun = 5;

class AlienAnimal {
public:
    AlienAnimal() { return; }

    int number_of_legs = 313;
};

class Predator : public AlienAnimal {
public:
    int number_of_fangs = 1703;

    void enable_hunting_mode() {
        number_of_legs += 57;
        number_of_fangs *= 2;
    }
};

Predator terrifying_creature;

int main() {

    terrifying_creature = Predator();
    terrifying_creature.enable_hunting_mode();

    return 0;
}
```

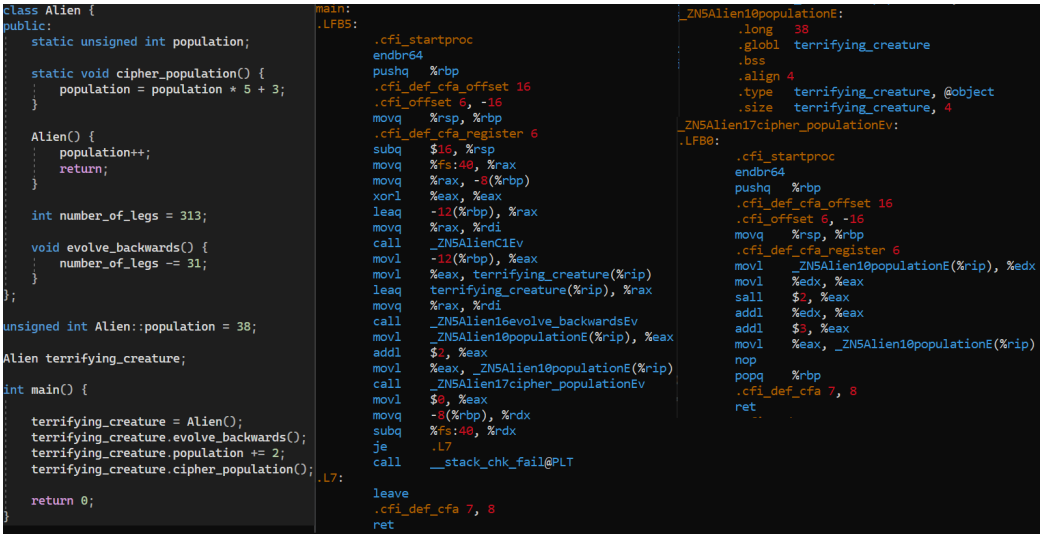
```
main:
.LFB7:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        subq    $16, %rsp
        movq    %fs:40, %rax
        movq    %rax, -8(%rbp)
        xorl    %eax, %eax
        movl    $0, -16(%rbp)
        movl    $0, -12(%rbp)
        leaq    -16(%rbp), %rax
        movq    %rax, %rdi
        call    _ZN8PredatorC1Ev
        movq    -16(%rbp), %rax
        movq    %rax, terrifying_creature(%rip)
        leaq    terrifying_creature(%rip), %rax
        movq    %rax, %rdi
        call    _ZN8Predator19enable_hunting_modeEv
        movl    $0, %eax
        movq    -8(%rbp), %rdx
```

```
_ZN8PredatorC2Ev:
.LFB5:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        subq    $16, %rsp
        movq    %rdi, -8(%rbp)
        movq    -8(%rbp), %rax
        movq    %rax, %rdi
        call    _ZN11AlienAnimalC2Ev
        movq    -8(%rbp), %rax
        movl    $1703, 4(%rax)
        nop
        leave
        .cfi_def_cfa 7, 8
        ret
```

```
_ZN11AlienAnimalC2Ev:
.LFB1:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movq    %rdi, -8(%rbp)
        movq    -8(%rbp), %rax
        movl    $313, (%rax)
        nop
        popq    %rbp
        .cfi_def_cfa 7, 8
        ret
```

7. При полиморфизме компилятор вставляет в листинг вызов метода нужного класса.

8. Работа со статиковыми полями происходит как с глобальными переменными. При вызове статикового метода указатель на экземпляр не запоминается в %rdi, поэтому метод никак не сможет получить доступ к экземпляру.



```
class Alien {
public:
    static unsigned int population;

    static void cipher_population() {
        population = population * 5 + 3;
    }

    Alien() {
        population++;
        return;
    }

    int number_of_legs = 313;

    void evolve_backwards() {
        number_of_legs -= 31;
    }
};

unsigned int Alien::population = 38;

Alien terrifying_creature;

int main() {

    terrifying_creature = Alien();
    terrifying_creature.evolve_backwards();
    terrifying_creature.population += 2;
    terrifying_creature.cipher_population();

    return 0;
}
```

```
main:
.LFB5:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        subq    $16, %rsp
        movq    %fs:40, %rax
        movq    %rax, -8(%rbp)
        xorl    %eax, %eax
        leaq    -12(%rbp), %rax
        movq    %rax, %rdi
        call    _ZN5AlienC1Ev
        movl    -12(%rbp), %eax
        movl    %eax, terrifying_creature(%rip)
        leaq    terrifying_creature(%rip), %rax
        movq    %rax, %rdi
        call    _ZN5Alien16evolve_backwardsEv
        movl    _ZN5Alien10populationE(%rip), %eax
        addl    $2, %eax
        movl    %eax, _ZN5Alien10populationE(%rip)
        call    _ZN5Alien17cipher_populationEv
        movl    $0, %eax
        movq    -8(%rbp), %rdx
        subq    %fs:40, %rdx
        je      .L7
        call    __stack_chk_fail@PLT
.L7:
        leave
        .cfi_def_cfa 7, 8
        ret
```

```
_ZN5Alien10populationE:
        .long   38
        .globl  terrifying_creature
        .bss
        .align 4
        .type   terrifying_creature, @object
        .size   terrifying_creature, 4
_ZN5Alien17cipher_populationEv:
.LFB0:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movl    _ZN5Alien10populationE(%rip), %edx
        movl    %edx, %eax
        sall    $2, %eax
        addl    %edx, %eax
        addl    $3, %eax
        movl    %eax, _ZN5Alien10populationE(%rip)
        nop
        popq    %rbp
        .cfi_def_cfa 7, 8
        ret
```

9. Перегрузка операторов реализуется очень просто. Создается функция, в название которой входят название класса и перегружаемого оператора (оператор умножения – ml, сложения – pl). Потом эта функция вызывается вместо перегруженного оператора.

10. Шаблоны работают максимально предсказуемо. Во время компиляции создаются листинги для каждой требуемой реализации.

```
template <typename T> class Alien {
public:
    T age;

    Alien();

    Alien(T age) {
        this->age = age;
    }

    void evolve_backwards(T amount_of_years_passed) {
        age += amount_of_years_passed;
    }
};

Alien<unsigned int> terrifying_creature = Alien(255U);
Alien<long long> horrifying_creature = Alien(9104300377LL);

int main() {

    terrifying_creature.evolve_backwards(127);
    horrifying_creature.evolve_backwards(-10000000000);

    return 0;
}
```

```
main:
.LFB2:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movl    $127, %esi
        leaq    terrifying_creature(%rip), %rax
        movq    %rax, %rdi
        call    _ZN5AlienIjE16evolve_backwardsEj
        movabsq $-10000000000, %rax
        movq    %rax, %rsi
        leaq    horrifying_creature(%rip), %rax
        movq    %rax, %rdi
        call    _ZN5AlienIxE16evolve_backwardsEx
        movl    $0, %eax
        popq    %rbp
        .cfi_def_cfa 7, 8
        ret
```

11. Enum в принципе работает предсказуемо

```cpp
#include <iostream>

int main() {
    enum Color { red = 2, green = 34, blue = 123 };
    Color r = green;

    switch (r)
    {
        case red:
            std::cout << "red\n";
            break;
        case green:
            std::cout << "green\n";
            break;
        case blue:
            std::cout << "blue\n";
            break;
    }
}
```

```asm
main:
.LFB1731:
        .cfi_startproc
        endbr64
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        subq    $16, %rsp
        movl    $34, -4(%rbp)
        movl    -4(%rbp), %eax
        cmpl    $123, %eax
        je      .L2
        cmpl    $123, %eax
        jg      .L3
        cmpl    $2, %eax
        je      .L4
        cmpl    $34, %eax
        je      .L5
        jmp     .L3
```