



# Co było do zrobienia

- Kompilacja stringów i wariantów
- Testowanie i bugfixy
- Rozdział pracy licencjackiej

- Dodanie obsługi typu `ptd::string`
- Implementacja kompilacji `own::var`

# Warianty

```
def main::typ() {  
    return own::var({  
        a => ptd::int(),  
        b =>  
            own::arr(ptd::int()),  
    });  
}  
# :a(1), :b([1,2,3])
```

```
struct main0type0type {  
    INT label;  
    union {  
        INT *a0case;  
        anon_type0ownarr00int  
            *b0case;  
        void *internal;  
    } value;  
    INT size;  
};
```

```
arr0int0push(&nl_arr_1, 1);  
arr0int0push(&nl_arr_1, 2);  
main0type0ov_mk(&nl_var_0, 1,  
                &nl_arr_1, sizeof);
```

```
var a : @main::type =  
      :b([1,2]);  
match (a) case :a(ref  
      x) {  
    x++;  
} case :b(ref x) {  
}
```

```
var a : @main::type =  
    :b([1,2]);  
match (a) case :a(ref  
    x) {  
    x++;  
} case :b(ref x) {  
}
```

```
arr0int0push(&nl_arr_1, 1);  
arr0int0push(&nl_arr_1, 2);  
main0type0ov_mk(&nl_var_0, 1,  
    &nl_arr_1, sizeof);  
nl_bool_4 = (nl_var_0.label == 0);  
if(nl_bool_4){ goto label_2;}  
nl_bool_4 = (nl_var_0.label == 1);  
if(nl_bool_4){ goto label_3;}  
nl_die();
```

```
var a : @main::type =  
    :b([1,2]);  
match (a) case :a(ref  
    x) {  
    x++;  
} case :b(ref x) {  
}
```

```
arr0int0push(&nl_arr_1, 1);  
arr0int0push(&nl_arr_1, 2);  
main0type0ov_mk(&nl_var_0, 1,  
    &nl_arr_1, sizeof);  
nl_bool_4 = (nl_var_0.label == 0);  
if(nl_bool_4){ goto label_2;}  
nl_bool_4 = (nl_var_0.label == 1);  
if(nl_bool_4){ goto label_3;}  
nl_die();  
label_2:  
nl_int_ptr_6 =  
    nl_var_0.value.a0case;  
nl_int_7 = 1;  
*nl_int_ptr_6 = *nl_int_ptr_6 +  
    nl_int_7;  
goto label_1;  
label_3: label_1:
```

# Plany na najbliższy tydzień

- **Rozdział pracy licencjackiej**
- Kompilacja hashy
- Zwalnianie pamięci