# Объявление объектов для таблиц (контейнеры и т.д.)

* Таблица констант (Номер константы, тип константы, значение).

class ConstTableItem{  
private:  
  
  
 ConstType\* constType;  
 string utf8;  
 double floatVal;  
 int val1, val2;  
  
public:  
 //методы  
};  
  
class ConstTable{  
private:  
 vector<ConstTableItem\*> items;  
public:  
 //методы на добалвения получение удаление, обновление  
};

* Таблица полей класса. Таблица хранит поля struct и константы, хранящиеся внутри trait.
* class FieldTableItem{  
  private:  
   TypeNode\* type;   
   bool isConst;   
   string id;   
   ExprNode\* value;   
  public:  
   //методы  
  };  
    
  class FieldTable{  
  private:  
   vector<FieldTableItem\*> items;  
  public:  
   //методы на добалвения получение удаление, обновление  
  };
* Таблица локальных переменных.

class VarTableItem{  
private:  
 TypeNode\* type;  
 bool isRef; // фактически это можно засунуть в тип после преобразования дерева  
 bool isMut;  
 string id;  
 ExprNode\* value;  
public:  
 //методы  
};  
  
class VarTable{  
private:  
 vector<VarTableItem\*> items;  
public:  
 //методы на добалвения получение удаление, обновление  
};

* Таблица методов. Свободные функции будут представлять собой статические методы анонимного класса с именем MY\_RUST\_ANONIMUS\_CLASS
* class MethodTableItem{  
  private:  
   string name;  
   TypeNode\* returnType;  
   ExprNode\* body;  
   MethodType\* methodType; // static, not\_static  
   FuncParamListNode\* params;  
   VarTable\* varTable;  
    
  public:  
   //методы  
  };  
    
  class MethodTable{  
  private:  
   string className;  
   map<string,MethodTableItem\*> items;  
  public:  
   //методы на добалвения получение удаление, обновление  
  };
* Таблица классов
* class ClassTableItem{  
  private:  
    
    
   ClassType classType; // stuct trait. можно разделить на классы  
   string name;  
   MethodTable\* methodTable;  
   FieldTable\* fieldTable;  
   ClassTableItem\* parent;  
    
  public:  
   //методы  
  };  
    
  class ConstTable{  
  private:  
   map<string,ClassTableItem\*> items;  
  public:  
   //методы на добалвения получение удаление, обновление  
  };

# Список ошибок (пока по докам)

*Borrow Checker Errors:*

error[E0502]: cannot borrow ... as mutable because it is also borrowed as immutable.

error[E0499]: cannot borrow ... as mutable more than once at a time.

error[E0308]: mismatched types.

*Ownership and Lifetimes:*

error[E0382]: use of moved value.

error[E0716]: temporary value dropped while borrowed.

error[E0495]: cannot infer an appropriate lifetime.

*Trait Bound Errors:*

error[E0277]: the trait bound ... is not satisfied.

error[E0308]: mismatched types.

*Pattern Matching Errors:*

error[E0004]: non-exhaustive patterns.

error[E0023]: unsure what fields are in this variant.

error[E0308]: mismatched types.

*Concurrency Errors:*

error[E0277]: the trait bound ... is not satisfied (related to Send and Sync traits).

*Unsafe Code Errors:*

error[E0133]: call to unsafe function is unsafe.

error[E0308]: mismatched types.

*Module and Crate Errors:*

error[E0432]: unresolved import.

error[E0463]: can't find crate for ....

*Macro Errors:*

error[E0451]: cannot continue macro expansion.

error[E0308]: mismatched types.

*Syntax Errors:*

error[E0010]: the type of this value must be known at this point.

error[E0282]: type annotations needed.