

F# Interactive в IDE Rider

Коёкин Ярослав

Группа 344

*Научный руководитель:
ст. препр. Я. А. Кириленко.*

*Консультант:
JetBrains, программист-инженер Е. П. Аудучинок.*

25 мая 2020 г.

Введение

REPL (от англ. read-eval-print loop) — форма организации интерактивной среды программирования в рамках средств интерфейса командной строки. В такой среде пользователь может вводить выражения, которые среда тут же будет вычислять, а результат вычисления отображать пользователю.

Введение

JetBrains Rider – кросс-платформенная интегрированная среда разработки программного обеспечения. Поддерживает проекты, основанные на платформах .Net Framework, .Net Core и Mono.



Постановка цели и задач

Расширение функциональных возможностей F# Interactive в IDE JetBrains Rider.

Постановка цели и задач

- ▶ Проанализировать существующую реализацию интерактивной консоли F#
- ▶ Реализовать в F# Interactive на уровне редактора:
 - ▶ систему автодополнений
 - ▶ подсветку кода
 - ▶ систему подсказок tooltips
 - ▶ систему навигации по коду
- ▶ Реализовать подсветку синтаксиса для вводимых команд и выводимых конструкций на уровне вывода истории команд
- ▶ Внести изменения в интерфейс

Аналоги. Ionide F# Interactive

Ionide – кросс-платформенное open-source расширение для Microsoft Visual Studio Code и Atom. Ionide предоставляет инструменты для поддержки языка F#. Также Ionide предоставляет интерактивную консоль.

Преимущества:

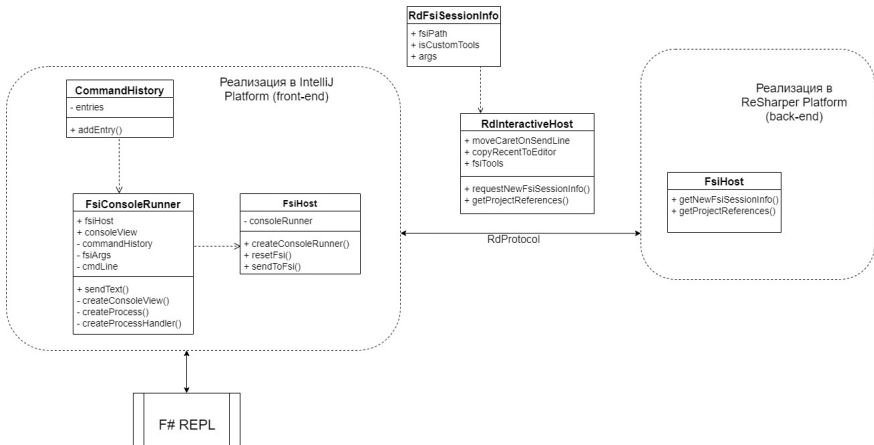
- ▶ Наличие автодополнения кода
- ▶ Подсветка кода
- ▶ Генерация скриптовых файлов

Недостатки:

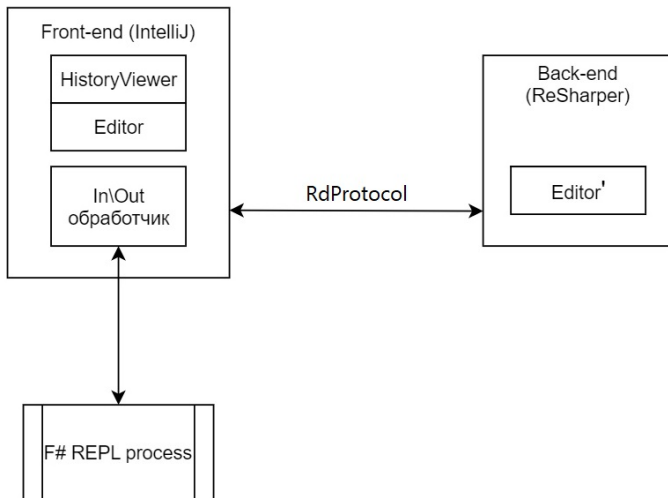
- ▶ Отсутствие системы подсказок
- ▶ Отсутствие системы навигации



F# Interactive в системе Rider



Реализация. Редактор кода.



Сложности, возникшие при реализации

F# Code

```
1 let testValue = 7
2 let testValue = 5
```

Untyped AST

```
1 ImplFile
2 (ParsedImplFileInput
3  ("Script.fsx",true,QualifiedFileNameOfFile Script$fsx,[],[]),
4  [SynModuleOrNamespace
5   ([Script],false,AnonModule,
6    [Let
7     (false,
8      [Binding
9       (None,NormalBinding,false,false,[],
10        PreXmlDoc ((1,13),FSharp.Compiler.Ast+XmlDocCollector),
11         SynValData
12          (None,SynValInfo ([],SynArgInfo ([],false,None)),None),
13         Named
14          (Wild Script.fsx (1,4--1,13) IsSynthetic=false,testValue,
15           false,None,Script.fsx (1,4--1,13) IsSynthetic=false),None),
16         Const (Int32 7,Script.fsx (1,16--1,17) IsSynthetic=false),
17         Script.fsx (1,4--1,13) IsSynthetic=false,
18         SequencePointAtBinding
19          Script.fsx (1,0--1,17) IsSynthetic=false)],
20         Script.fsx (1,0--1,17) IsSynthetic=false);
21     Let
22      (false,
23       [Binding
24        (None,NormalBinding,false,false,[],
25         PreXmlDoc ((2,13),FSharp.Compiler.Ast+XmlDocCollector),
26         SynValData
27          (None,SynValInfo ([],SynArgInfo ([],false,None)),None),
28         Named
29          (Wild Script.fsx (2,4--2,13) IsSynthetic=false,testValue,
30           false,None,Script.fsx (2,4--2,13) IsSynthetic=false),None),
31         Const (Int32 5,Script.fsx (2,16--2,17) IsSynthetic=false),
32         Script.fsx (2,4--2,13) IsSynthetic=false,
33         SequencePointAtBinding
34          Script.fsx (2,0--2,17) IsSynthetic=false)],
35         Script.fsx (2,0--2,17) IsSynthetic=false)],PreXmlDocEmpty,[],None),
36     Script.fsx (1,0--2,17) IsSynthetic=false)],(true,true))
```

Решение

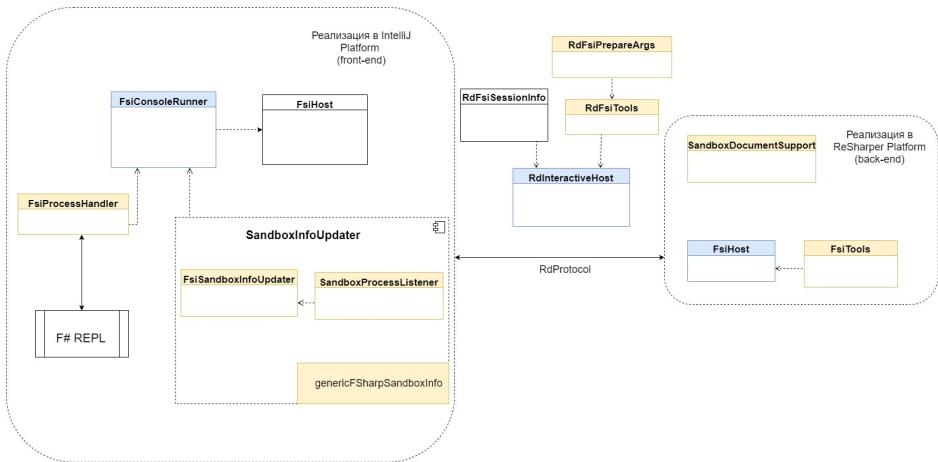
F# Code

```
1  [<AutoOpen>]  
2  module autoOpenModule =  
3  ... let testValue = 7  
4  let testValue = 5
```

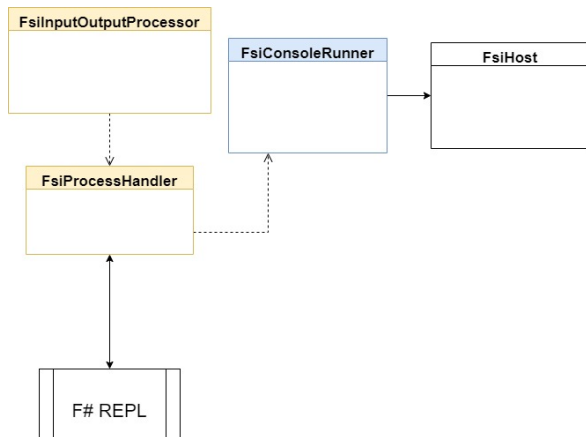
Untyped AST

```
1 ImplFile  
2 (ParsedImplFileInput  
3 ("Script.fsx",true,QualifiedModuleNameOfFile Script$fsx,[],[]),  
4 [SynModuleOrNamespace  
5 ([Script],false,AnonModule,  
6 [DoExpr  
7 (SequencePointAtBinding Script.fsx (1,0--1,10) IsSynthetic=false,  
8 ArrayOrListOfSeqExpr  
9 (false,  
10 CompExpr  
11 (true,( contents = true ),Ident AutoOpen,  
12 Script.fsx (1,1--1,9) IsSynthetic=false),  
13 Script.fsx (1,0--1,10) IsSynthetic=false),  
14 Script.fsx (1,0--1,10) IsSynthetic=false);  
15  
16 NestedModule  
17 (ComponentInfo  
18 ([],[],[autoOpenModule],  
19 PreXmlDoc ((2,6),FSharp.Compiler.Ast+XmlDocCollector),false,  
20 None,Script.fsx (2,0--2,21) IsSynthetic=false),false,  
21 [Let  
22 (false,  
23 [Binding  
24 (None,NormalBinding,false,false,[],  
25 PreXmlDoc ((3,17),FSharp.Compiler.Ast+XmlDocCollector),  
26 SynValData  
27 (None,SynValInfo ([],SynArgInfo ([],false,None)),None),  
28 Named  
29 (Wild Script.fsx (3,8--3,17) IsSynthetic=false,testValue,  
30 false,None,Script.fsx (3,8--3,17) IsSynthetic=false),  
31 None,  
32 Const (Int32 7,Script.fsx (3,20--3,21) IsSynthetic=false),  
33 Script.fsx (3,8--3,17) IsSynthetic=false,  
34 SequencePointAtBinding  
35 Script.fsx (3,4--3,21) IsSynthetic=false)],  
36 Script.fsx (3,4--3,21) IsSynthetic=false)],false,  
37 Script.fsx (2,0--3,21) IsSynthetic=false);  
38 Let  
39 (false,  
40 [Binding  
41 (None,NormalBinding,false,false,[],  
42 PreXmlDoc ((4,13),FSharp.Compiler.Ast+XmlDocCollector),  
43 SynValData  
44 (None,SynValInfo ([],SynArgInfo ([],false,None)),None),  
45 Named  
46 (Wild Script.fsx (4,4--4,13) IsSynthetic=false,testValue,  
47 false,None,Script.fsx (4,4--4,13) IsSynthetic=false),None,  
48 Const (Int32 5,Script.fsx (4,16--4,17) IsSynthetic=false),  
49 Script.fsx (4,4--4,13) IsSynthetic=false,  
50 SequencePointAtBinding  
51 Script.fsx (4,0--4,17) IsSynthetic=false)],PreXmlDocEmpty,[],None,  
52 Script.fsx (1,0--4,17) IsSynthetic=false)],(true, true))
```

Реализация. Архитектура решения задачи



Реализация. Подсветка синтаксиса



Итоги

- ▶ Проанализирована существующая реализация интерактивной консоли F#
- ▶ Реализованы в F# Interactive на уровне редактора:
 - ▶ система автодополнений
 - ▶ подсветка кода
 - ▶ система подсказок tooltips
 - ▶ система навигации по коду
- ▶ Реализована подсветка синтаксиса для вводимых команд и выводимых конструкций на уровне вывода истории команд.
- ▶ Внесены изменения в интерфейс

Код (форк): <https://github.com/yarkoe/fsharp-support>