

Весна 2015

Катя Тузова
JetBrains

Интеграция CheckIO и PyCharm

CheckIO

<http://checkio.org> – онлайн-сервис для обучения языку Python в игровой форме.

Плюсы:

- Игровой формат обучения
- Большая база задач

Минусы:

- Очень упрощенный редактор кода, в котором сложно писать код
- Студенту приходится копировать код задачи и редактировать его в более продвинутом редакторе, а потом копировать решение в браузер.

CheckIO



CheckIO

The screenshot shows the CheckIO website interface for the 'Xs and Os Referee' mission. The browser address bar shows the URL `www.checkio.org/mission/x-o-referee/solve/`. The Python code editor contains the following code:

```
1 def checkio(game_result):
2     return "D" or "X" or "O"
3
4 if __name__ == '__main__':
5     #These "asserts" using only for self-checking and not necessary for auto-
6     assert checkio([
7         "X.O",
8         "XX.",
9         "XOO"]) == "X", "Xs wins"
10    assert checkio([
11        "OO.",
12        "XOX",
13        "XOX"]) == "O", "Os wins"
14    assert checkio([
15        "OOX",
16        "XXO",
17        "OXX"]) == "D", "Draw"
18    assert checkio([
19        "O.X",
20        "XX.",
21        "XOO"]) == "X", "Xs wins again"
22
23
```

The 'Check results' section explains the game and the task:

Tic-Tac-Toe, sometimes also known as Xs and Os, is a game for two players (X and O) who take turns marking the spaces in a 3x3 grid. The player who succeeds in placing three respective marks in a horizontal, vertical, or diagonal rows (NW-SE and NE-SW) wins the game. But we will not be playing this game. You will be the referee for this games results. You are given a result of a game and you must determine if the game ends in a win or a draw as well as who will be the winner. Make sure to return "X" if the X-player wins and "O" if the O-player wins. If the game is a draw, return "D".

Three Tic-Tac-Toe grids are shown with their corresponding code and results:

<code>checkio(["X.O", "XX.", "XOO"])</code>	<code>checkio(["OO.", "XOX", "XOX"])</code>	<code>checkio(["OOX", "XXO", "OXX"])</code>
Result: "X"	Result: "O"	Result: "D"

The 'Output (Python console)' section shows a Tic-Tac-Toe grid and buttons for 'Random' and 'Check':

Buttons: Random, Check

The 'Precondition' section states: `len(game_result) == 3` and `all(len(row) == 3 for row in game_result)`. A blue button at the bottom says "I have no idea how to start solving this mission".

PyCharm EDU

Образовательная версия PyCharm.

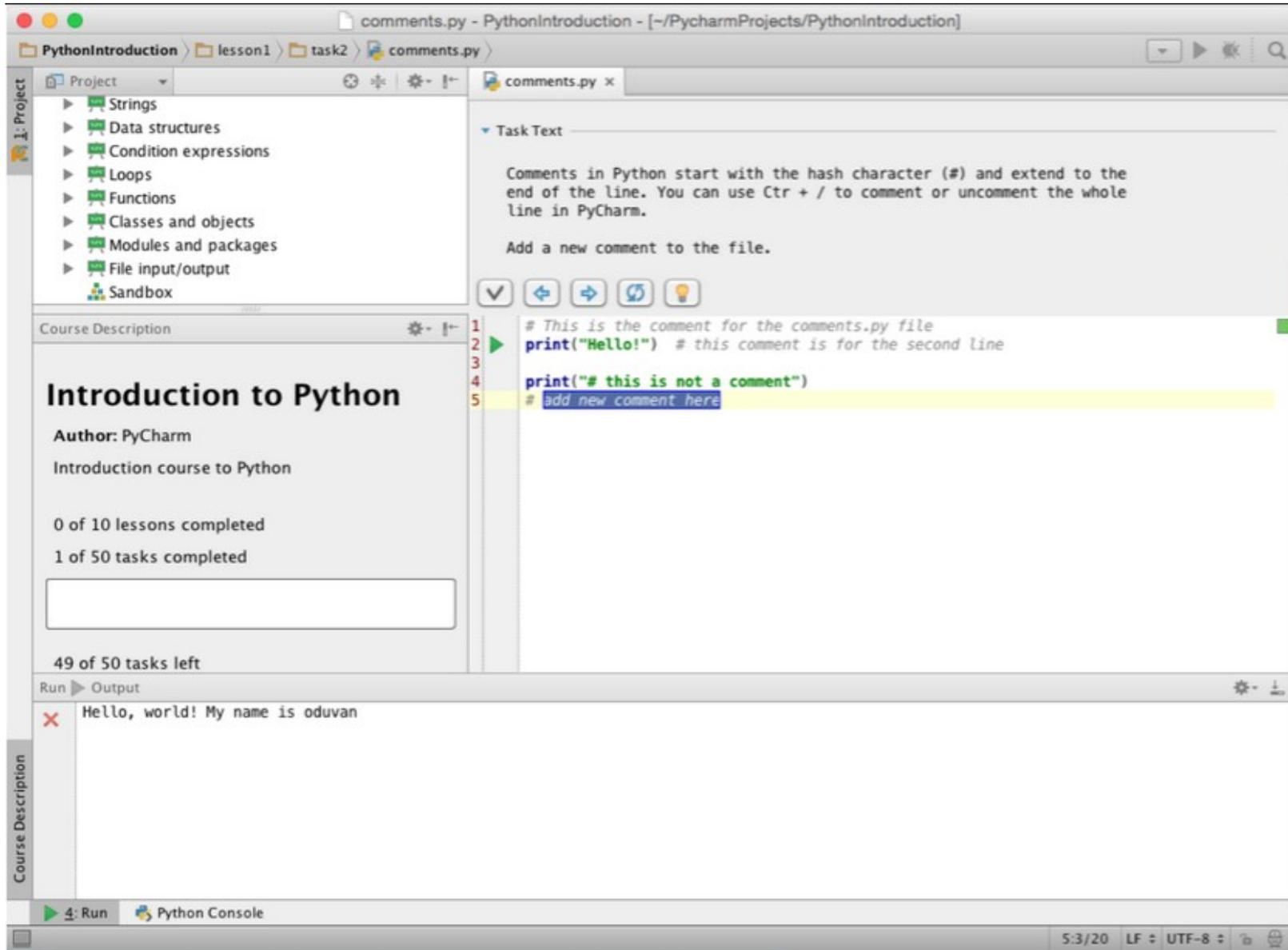
Плюсы:

- Есть все возможности настоящей IDE по редактированию кода.

Минусы:

- Не хватает геймификации.

PyCharm EDU



CheckIO и PyCharm

Задача:

- Совместить два подхода

Формат:

- 1 человек
- Место – JetBrains
- Желательно знать Java

Визуальный дебаггер для языка R

R Debugger

```
r-generator.r x
1  args <- commandArgs(TRUE)
2  packageNames <- .packages(all = TRUE)
3  searchPath <- search()
4
5  is.identifier <- function(str) {
6    return(grepl("^[[:alpha:]]|_|\\.|[[:alpha:]]|[[:digit:]]|_|\\.|)*$")
7  }
8
9
10
11  for (name in packageNames) {
12    # if (name == "base") next
13    shouldLoadLibrary = FALSE
14    pName = paste("package", name, sep=":")
15    if (!pName %in% searchPath)
16      shouldLoadLibrary = TRUE
17    if (shouldLoadLibrary) {
18      library(package=name, character.only=TRUE)
19    }
20
21    symbolList <- ls(pName)
22
23    dirName = paste(args[1], name, sep="/")
24    dir.create(dirName)
25
26    for(symbol in symbolList) {
27      obj <- get(symbol)
28      fileName <- paste(paste(dirName, symbol, sep="/"), "r", sep="")
29    }
30  }
```

R Debugger

Задача:

- Изучить существующие решения (Rstudio и StatET)
- Написать свой визуальный дебаггер для R как плагин к IDEA

Формат:

- 1 человек
- Место – JetBrains
- Желательно знать Java, R

Интерактивные графики в PyCharm

Интерактивные графики

Задача:

- Tool window с графиками, полученными в результате выполнения python скрипта.
- Добавить интерактивность графиков
- Расширить применение на R.

Формат:

- 1 человек
- Место – JetBrains
- Желательно знать Java

Существующие решения

- Интерактивные виджеты в ipython notebook [1]
- Пакет manipulate для R [2]

[1] <http://nbviewer.ipython.org/github/adrn/ipython/blob/master/examples/Interactive%20Widgets/Index.ipynb>

[2] <https://support.rstudio.com/hc/en-us/articles/200551906-Interactive-Plotting-with-Manipulate>