

IDE от JetBrains PyCharm

Обзор функций разработки ПО

Авторы: студенты группы ИВТ-2.2

- Магер Егор Владимирович
- Спирянов Максим Дмитриевич
- Сурков Андрей Анатольевич
- Киняев Илья Константинович

Коротко о PyCharm

PyCharm - это кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IDE IntelliJ IDEA. Предоставляет пользователю комплекс средств для написания кода и визуальный отладчик.



JetBrains PyCharm

Полноценная Среда Разработки!

Кроссплатформенность



Поддержка всех современных ОС

Code Inspection



Исправление ошибок

```
# Find force at the contact patch
F_wheel = self.get_tire_force(self.tire_width, F_wheel: 15089.592295922961
                             self.tire_aspect,
                             self.rim_size,
                             T_wheel)

# Brake force
F_brake = (self.braking / 100) * self.brake_force F_brake: 0.0

# Resultant force is: + F_engine - F_air - F_brake, where positive is forward
force = F_wheel - F_air - F_brake - F_roll force: 14769.253214784112

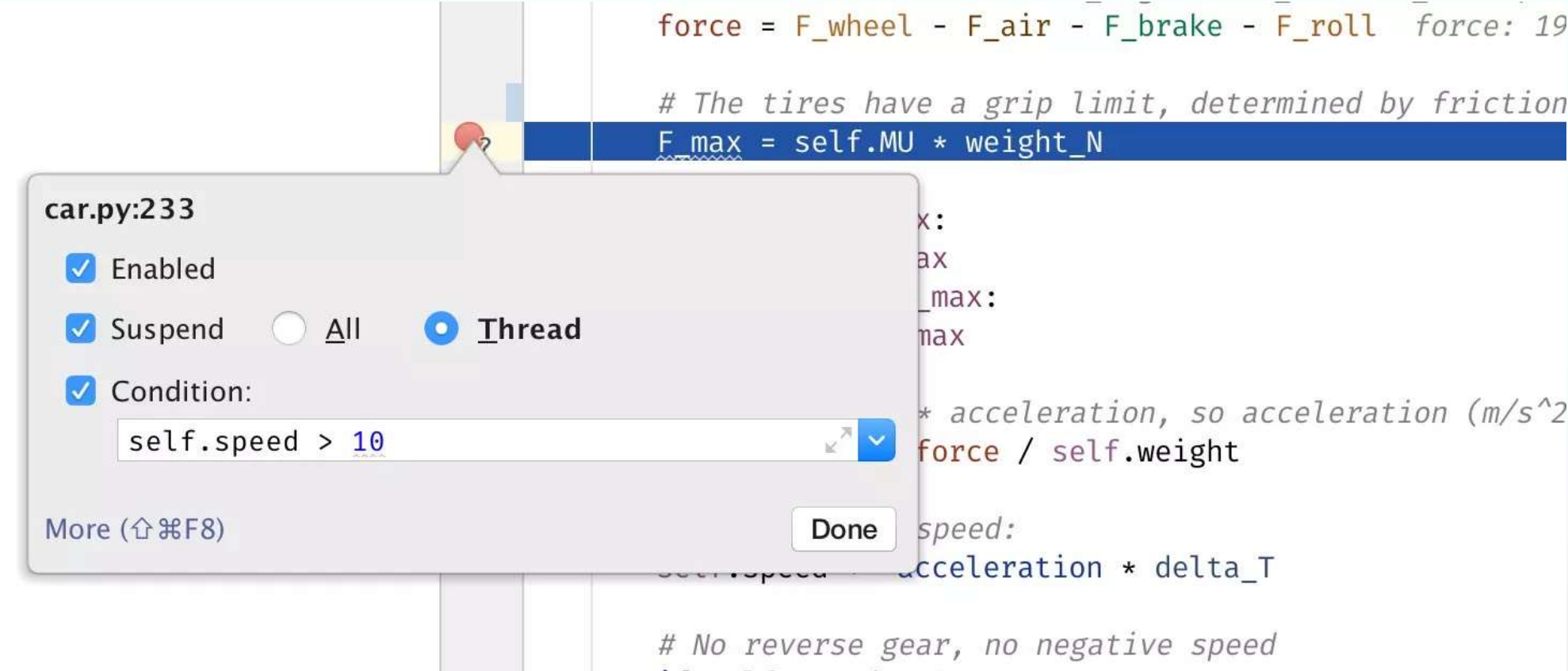
# The tires have a grip limit, determined by friction and (aerodynamic) weight
F_max = self.MU * weight_N F_max: 15362.460000000001

if force > F_max:
    force = F_max
elif force < -F_max:
    force = -F_max

# Force = mass * acceleration, so acceleration (m/s^2) = force (N) / mass (kg)
acceleration = force / self.weight acceleration: 8.488076560220755

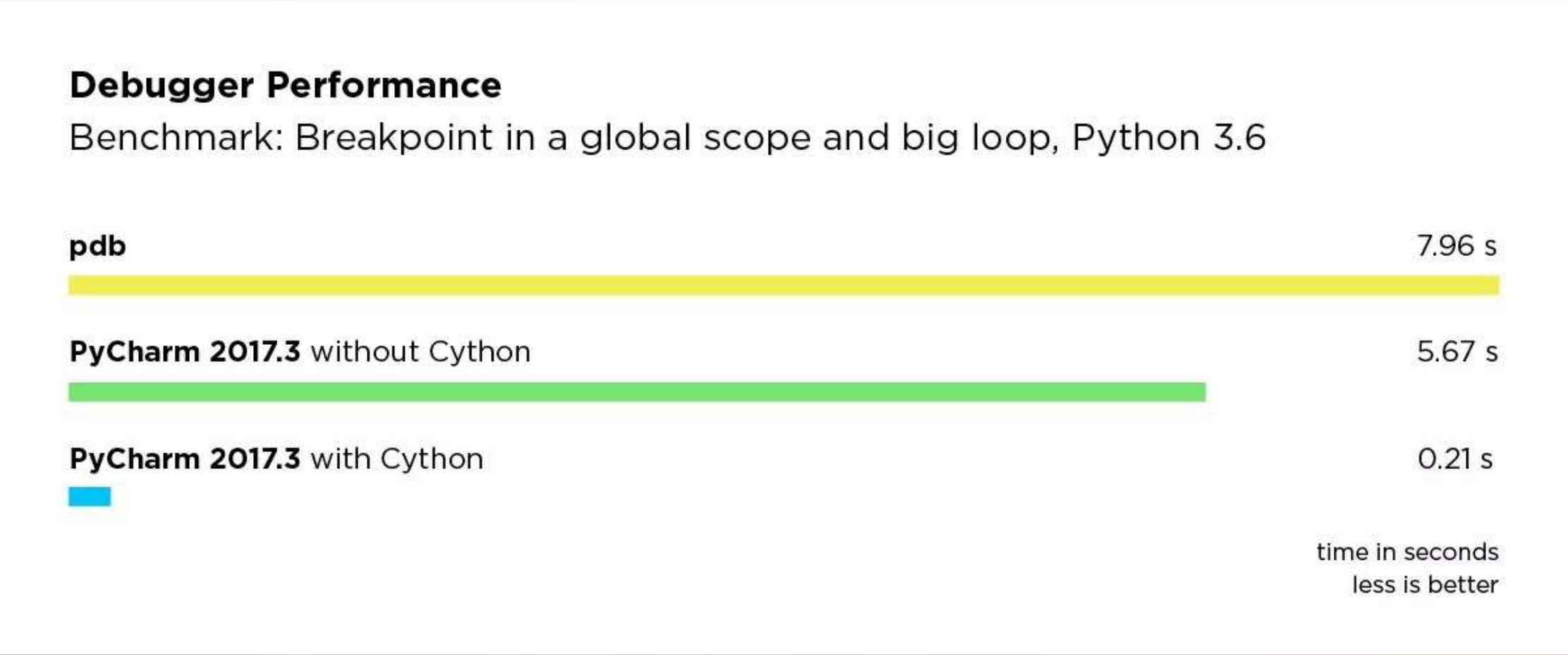
# The cars new speed:
self.speed += acceleration * delta_T
```

Просмотр значения переменных

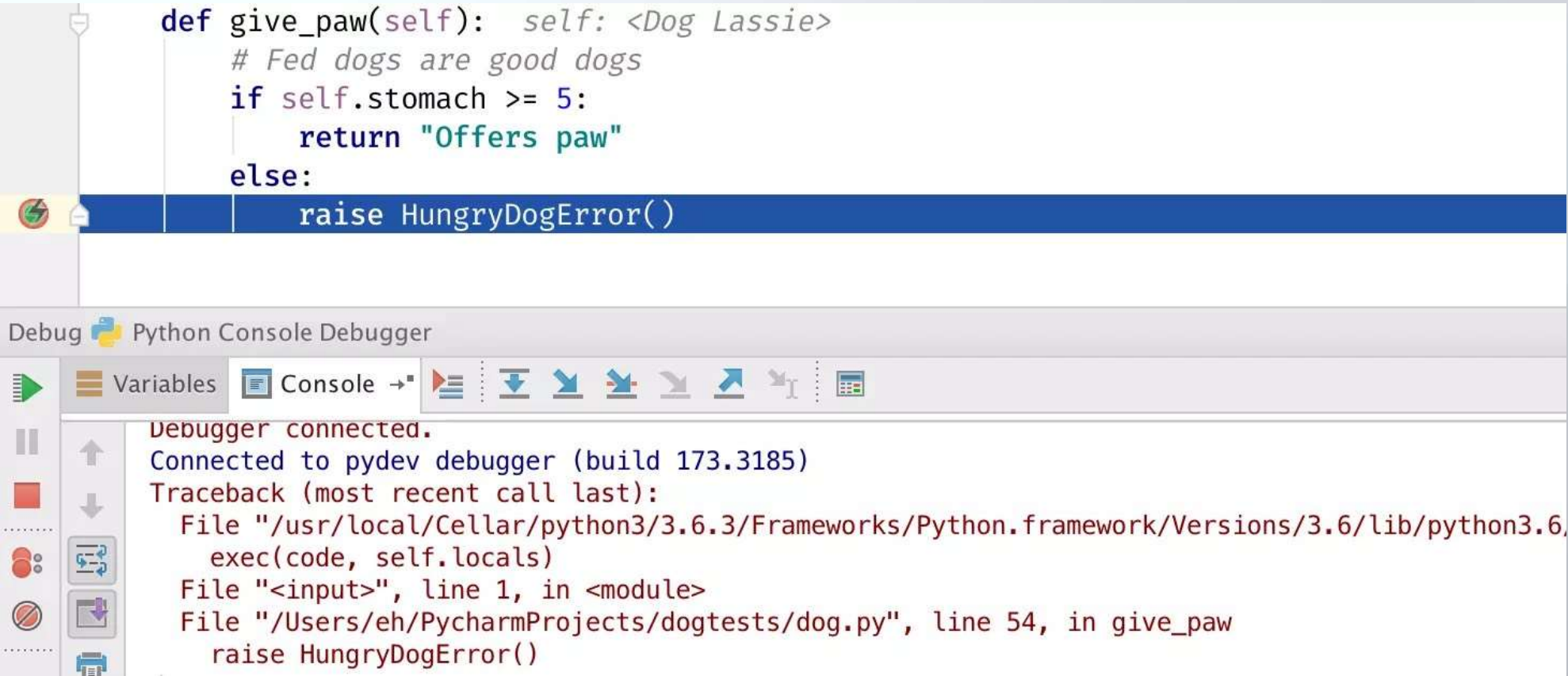


Гибкие точки останова

Отладчик



Скорость



Определение исключений

Гибкие точки останова

```
force = F_wheel - F_air - F_brake - F_roll force: 19
```

```
# The tires have a grip limit, determined by friction
```

```
F_max = self.MU * weight_N
```

car.py:233

☒ Enabled

☒ Suspend ☐ All ☒ Thread

☒ Condition:

self.speed > 10

More (⇧⌘F8)

Done

x:

ax

_max:

max

* acceleration, so acceleration (m/s^2

force / self.weight

speed:

acceleration * delta_T

```
# No reverse gear, no negative speed
```

Просмотр значения переменных

```
# Find force at the contact patch
F_wheel = self.get_tire_force(self.tire_width, F_wheel: 15089.592295922961
                             self.tire_aspect,
                             self.rim_size,
                             T_wheel)

# Brake force
F_brake = (self.braking / 100) * self.brake_force F_brake: 0.0

# Resultant force is: + F_engine - F_air - F_brake, where positive is forward
force = F_wheel - F_air - F_brake - F_roll force: 14769.253214784112

# The tires have a grip limit, determined by friction and (aerodynamic) weight
F_max = self.MU * weight_N F_max: 15362.460000000001

if force > F_max:
    force = F_max
elif force < -F_max:
    force = -F_max

# Force = mass * acceleration, so acceleration (m/s^2) = force (N) / mass (kg)
acceleration = force / self.weight acceleration: 8.488076560220755

# The cars new speed:
self.speed += acceleration * delta_T
```

Определение исключений

```
def give_paw(self):  self: <Dog Lassie>
    # Fed dogs are good dogs
    if self.stomach >= 5:
        return "Offers paw"
    else:
        raise HungryDogError()
```

Debug Python Console Debugger

Variables Console →

Debugger connected.

Connected to pydev debugger (build 173.3185)

Traceback (most recent call last):

File "/usr/local/Cellar/python3/3.6.3/Frameworks/Python.framework/Versions/3.6/lib/python3.6.
exec(code, self.locals)

File "<input>", line 1, in <module>

File "/Users/eh/PycharmProjects/dogtests/dog.py", line 54, in give_paw
raise HungryDogError()

Скорость

Debugger Performance

Benchmark: Breakpoint in a global scope and big loop, Python 3.6

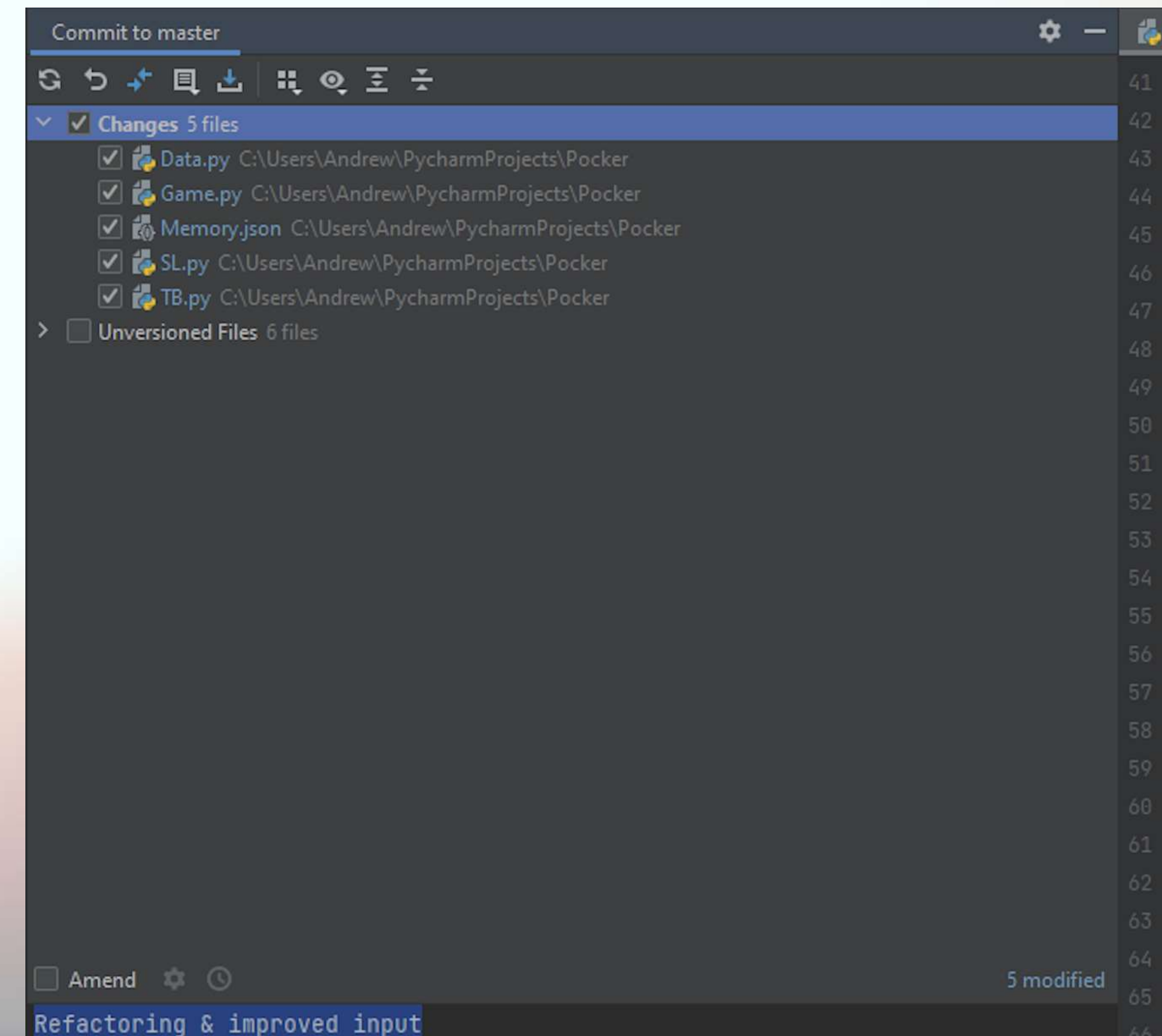


time in seconds
less is better

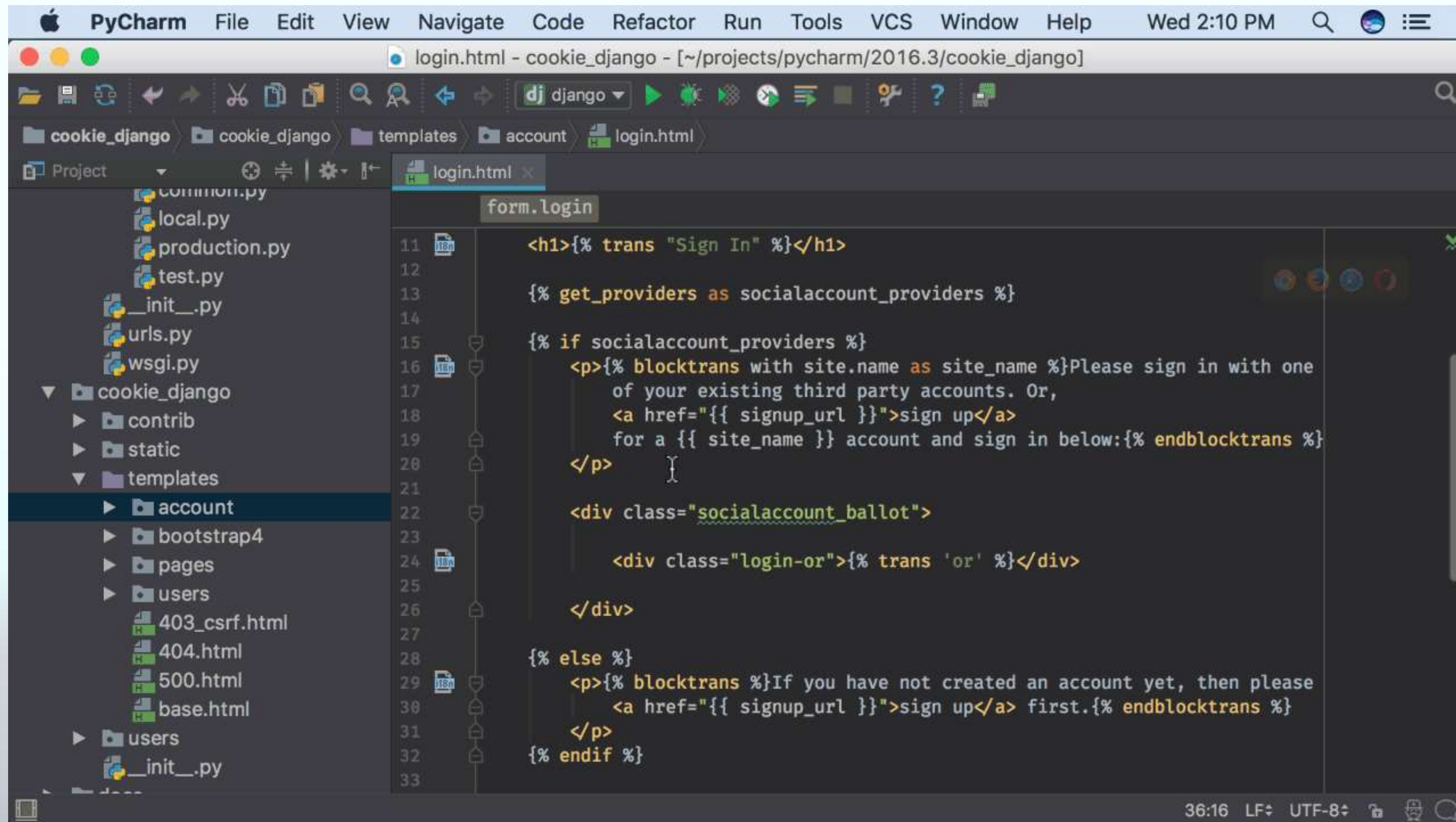
Интеграция с системами контроля версий

```
59         break
60     elif not(int(change[i]) in bones):
61
62     elif not(z[i] == '1' or z[i] == '2' or z[i] == '3' or z[i] == '4' or z[i] == '5' or z[i] == '6'):
```

Интеграция с Git, Mercurial и др.



Поддержка различных фреймворков и библиотек



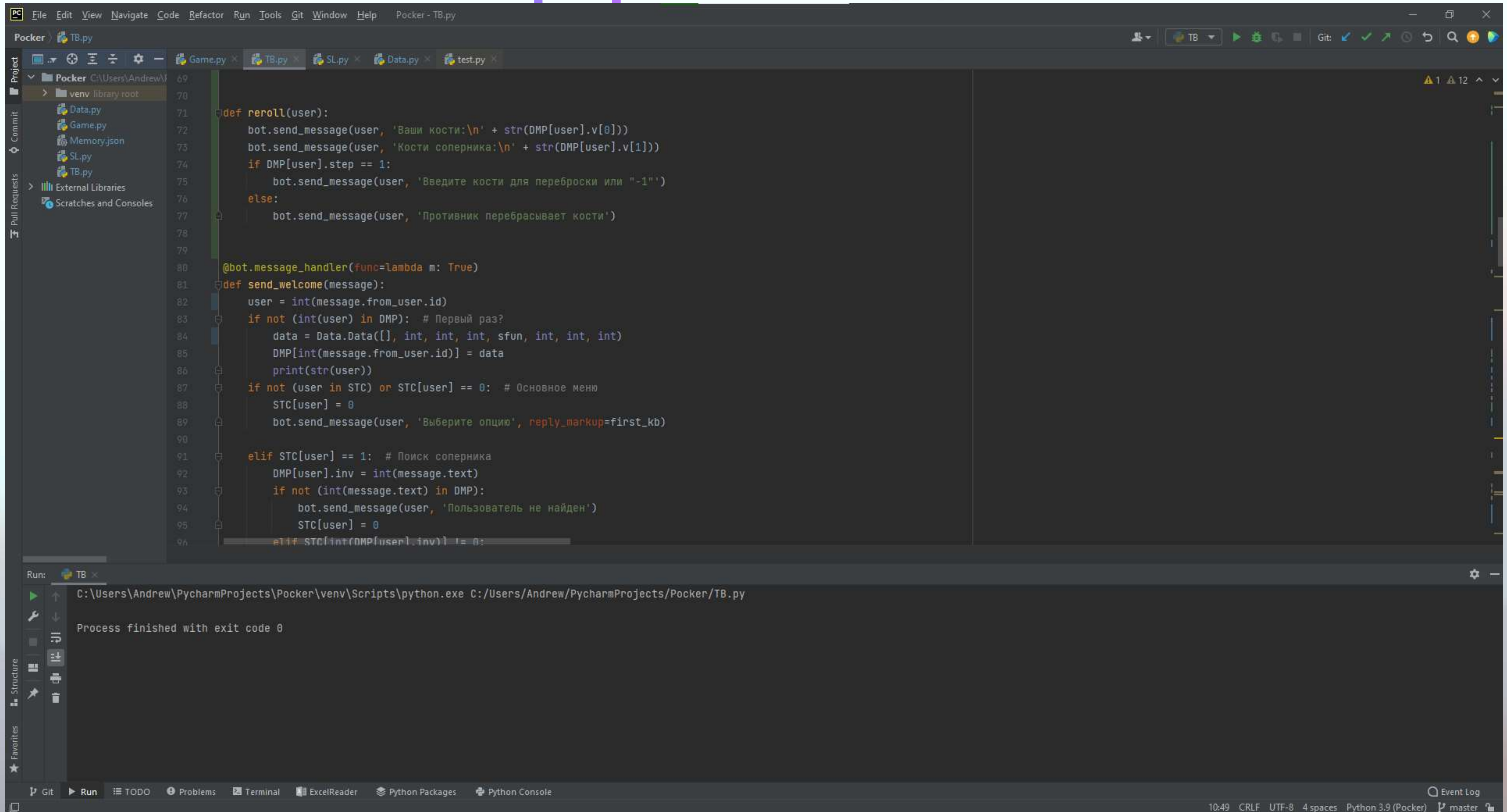
Создание проекта Django

Другие Функции

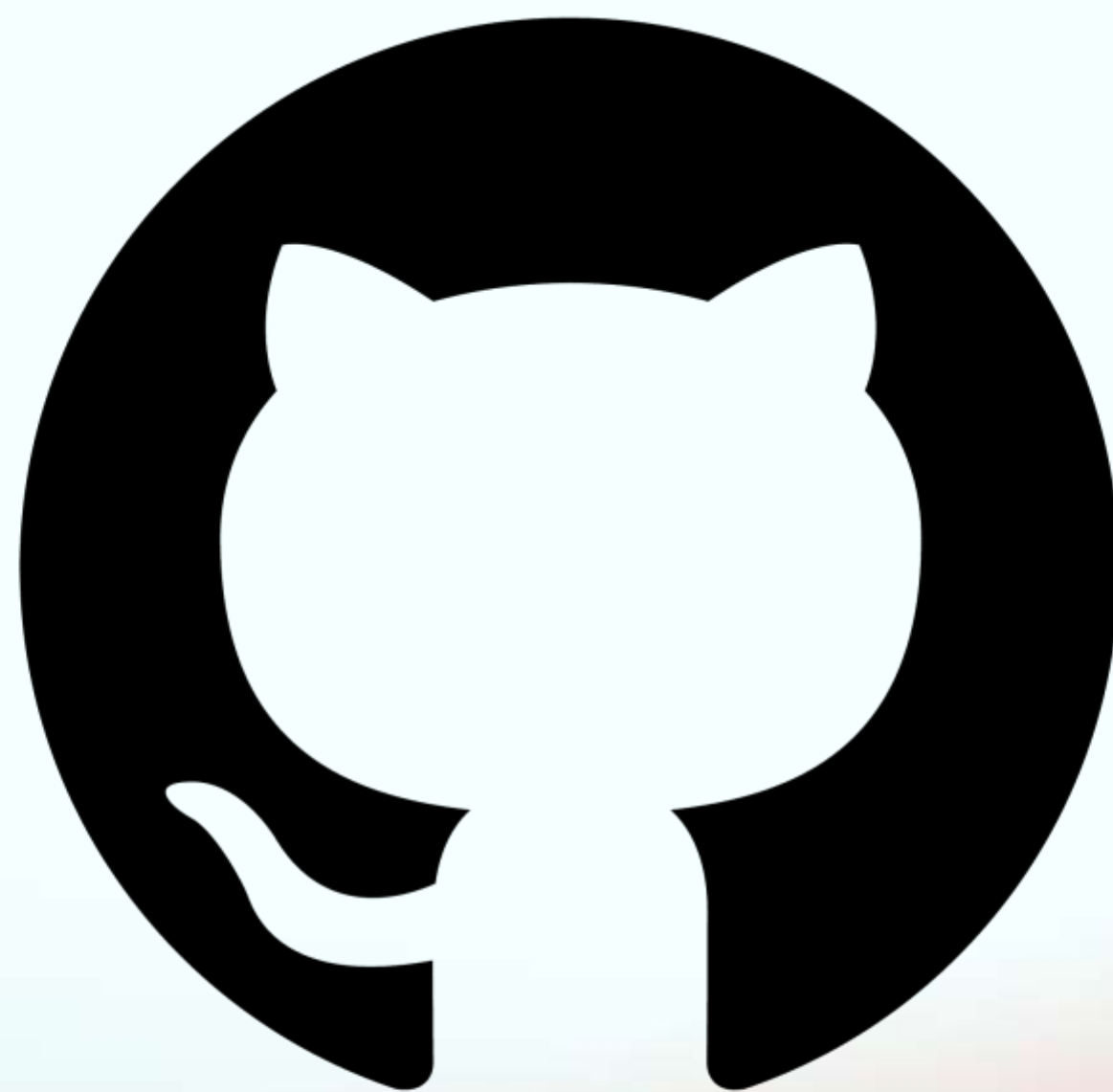
Горячие Клавиши

- ⦿ **Tab:** Автодополнение кода
- ⦿ **Ctrl+.: “Свернуть”** блок кода
- ⦿ **Ctrl + B:** Перейти к определению переменной или функции.
- ⦿ **Ctrl + /:** Закомментировать или раскомментировать выделенный текст.
- ⦿ **Alt + Enter:** Открыть контекстное меню с предложениями по исправлению кода.
- ⦿ **Ctrl + Shift + T:** Создать тест для класса или функции.
- ⦿ **Ctrl + Shift + A:** Поиск действий и настроек PyCharm.
- ⦿ **Ctrl + E:** Открыть список последних файлов.
- ⦿ **Alt + F7:** Поиск использований переменной или метода.
- ⦿ **Ctrl + Alt + L:** Форматирование выделенного кода.
- ⦿ **Shift + F10:** Запустить программу.
- ⦿ **Shift + F9:** Запустить тесты в проекте.

Интерфейс и дизайн



Интеграция с GitHub



GitHub

Возможности GitHub

Возможности GitHub:

- ✓ Инициализация репозитория
- ✓ Клонирование репозитория
- ✓ Коммит и пуш
- ✓ Просмотр изменений
- ✓ Управление ветками
- ✓ Работа с Issues и Pull Requests

Нагрузка на систему

Иногда PyCharm может создавать высокую нагрузку на ПК из-за:

1. Больших проектов: Много файлов и зависимостей.
2. Анализа кода: Проверки синтаксиса и подсветки кода.
3. Отладки: Запуска отладчика и отслеживания выполнения кода.
4. Виртуальных окружений: Дополнительные ресурсы для каждого окружения.
5. Веб-приложений: Использование встроенных серверов.
6. Баз данных и сервисов: Обращения к ним.
7. Плагинов и расширений: Множества установленных плагинов.

Спасибо за внимание!