Министерство образования и науки российской федерации  
федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Факультет информационных технологий и программирования

Кафедра информационных систем

Лабораторная работа № 1

Инструментальные средства разработки ПО

Настройка IDE

Выполнил студент группы № M3306:

Назаров Д.А.

Проверил:

Липкин Евгений Олегович

Санкт-Петербург

2018

**PyCharm**

PyCharm — [интегрированная среда разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) для языка программирования [Python](https://ru.wikipedia.org/wiki/Python" \o "Python). Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на [Django](https://ru.wikipedia.org/wiki/Django" \o "Django). PyCharm разработана компанией [JetBrains](https://ru.wikipedia.org/wiki/JetBrains) на основе [IntelliJ IDEA](https://ru.wikipedia.org/wiki/IntelliJ_IDEA" \o "IntelliJ IDEA).

**Навигация по проекту**

Левая часть основного окна отражает структуру проекта, показывая директории и исполняемые файлы проекта. Чуть ниже располагается вкладка, в которой можно посмотреть структуру кода – классы, методы и т.д. Сверху окна располагается панель инструментов.

Центральную часть занимает, собственно, сам редактор кода, причём можно открыть несколько вкладок, удобно их расположить и редактировать код сразу в нескольких файлах. Также есть возможность сворачивать отдельные куски кода, расставлять брейкпойнты в левой части редактора кода, что может быть полезно при отладке программы.

Внизу располагается вывод, сама консоль Pycharma, список переменных с текущими значениями, что также полезно при отладке.

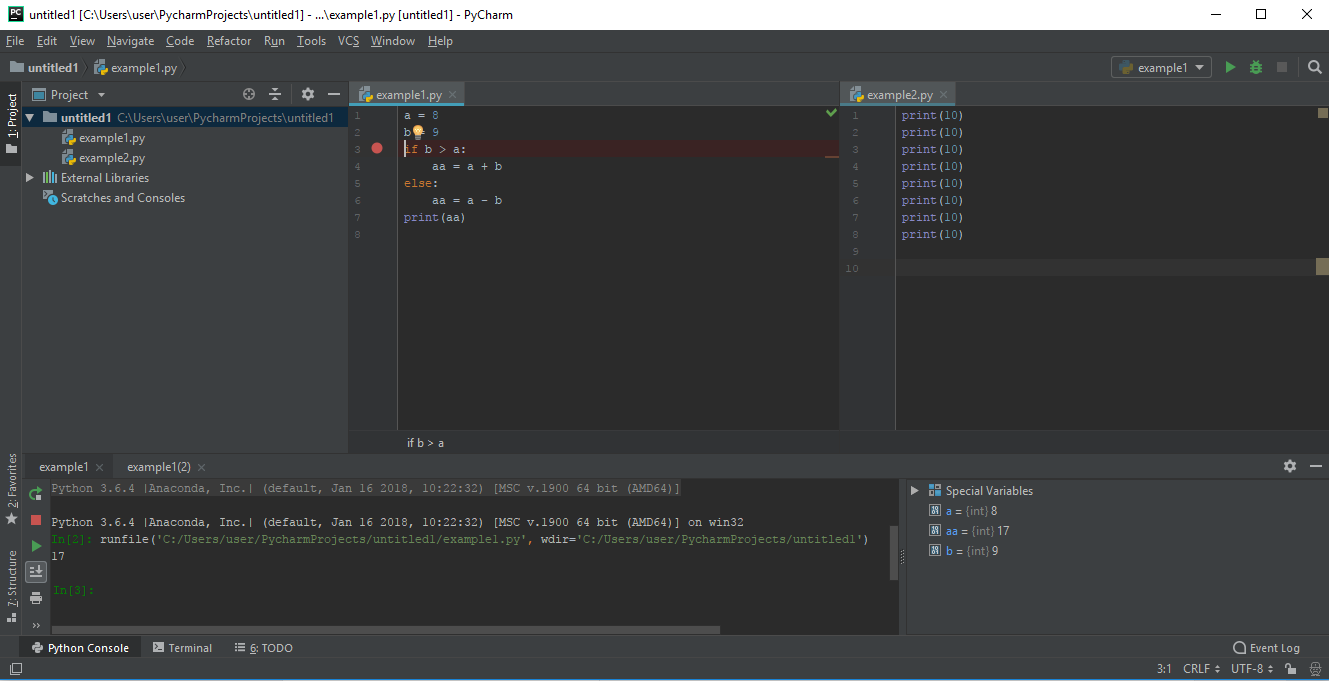


Рис. 1 Основное окно

**Лаунчеры**

**Линтинг**

В Pycharme очень быстрый и функциональный анализатор кода. Все ошибки подсвечиваются и подчёркиваются ещё на этапе написания. Анализатор кода обращает внимание на неправильный синтаксис, неиспользуемые переменные или целые куски кода, ошибки, запрещённые методы, просто неэффективный код, а также предлагает внести корректировки.

Приоритеты инспекшнов в анализаторе кода легко настраиваются вручную. Различаются они следующим образом: красные - ошибки , жёлтые предупреждения, слабо жёлтые – слабое предупреждение и т.д.

В настройках на вкладке Incpections можно изменять приоритеты различных инспекшенов «на свой лад».

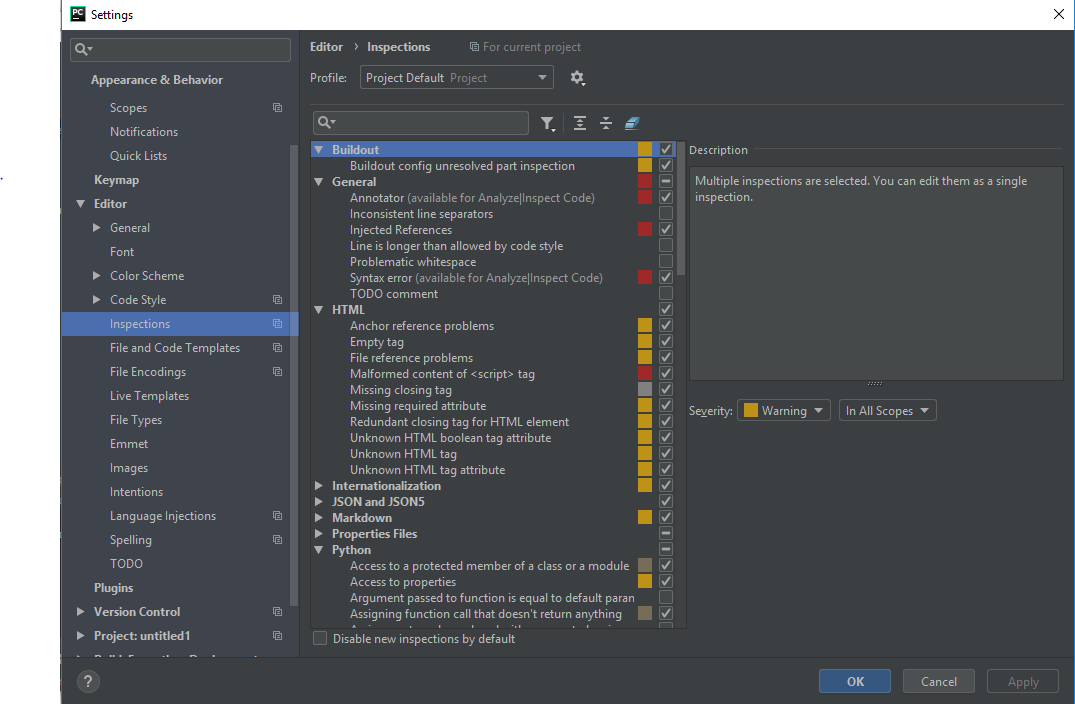
+

Рис. 2 Настройка инспекшенов

**Интерпретатор**

При создании нового проекта в PyCharme предлагается выбрать один из тех, которые он поддерживает. Это либо обычный системный интерпретатор, либо Virtualenv среды: Virtualenv, Pipenv, Conda.

[Virtualenv](https://pypi.python.org/pypi/virtualenv) является очень популярным инструментом, который создает изолированные среды Python для библиотек Python.

Pipenv предназначен для объединения Pipfile, pip и virtualenv в одну команду в командной строке.

Conda - менеджер пакетов и система управления окружающей средой.

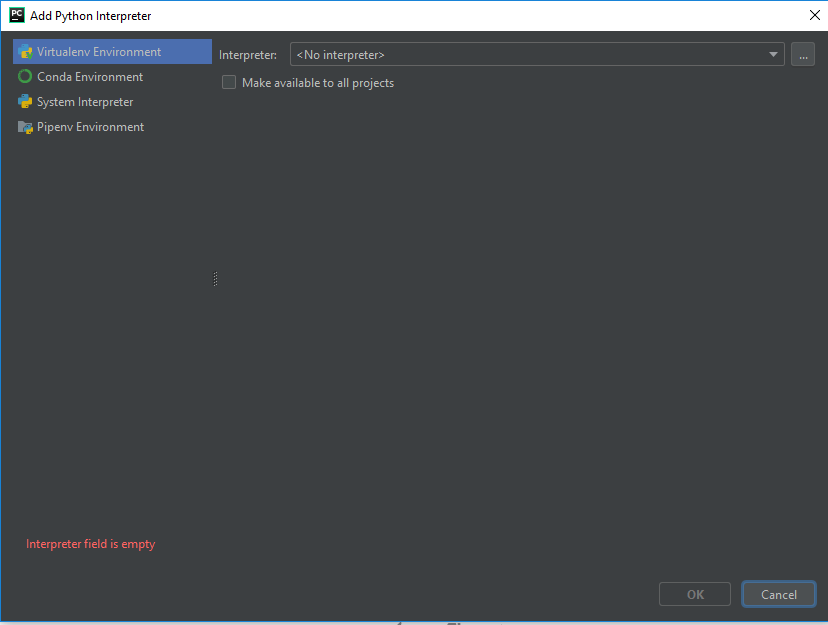


Рис. 3 Выбор интерпертатора 1

Также интерпретатор можно выбрать в настройках на вкладке Project Interpretetor, где так же можно увидеть список установленных пакетов:

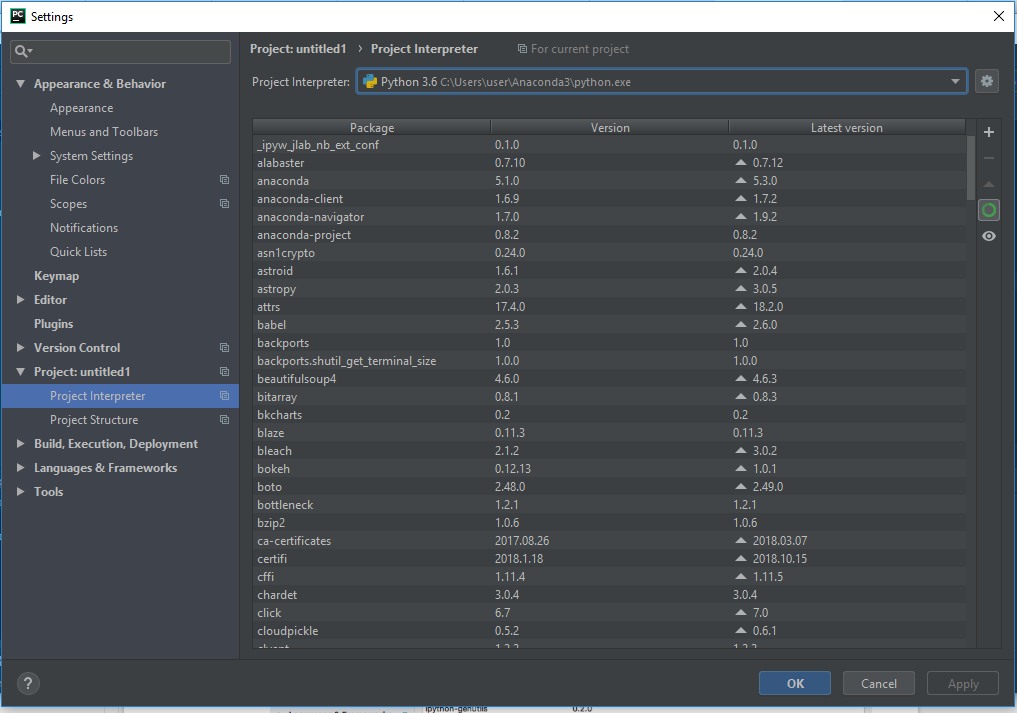


Рис. 4 Выбор интерпретатора 2

Установить и подключить какие-либо библиотеки можно в консоли при помощи команды pip install.

**Тестирование**

В Pycharme существует также возможность для юнит-тестирования.

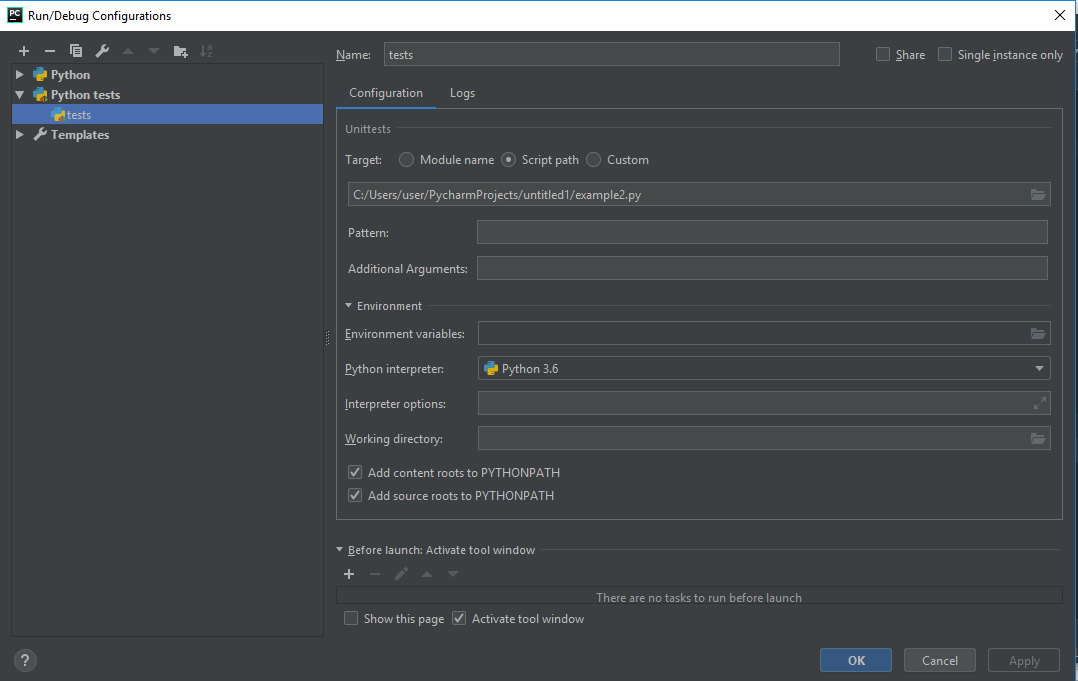


Рис. 5 Создание теста

**Горячие клавиши:**

***Работа с закладками:***

|  |  |
| --- | --- |
| Поставить или снять закладку | F11 |
| Аналогично с присвоением буквы или цифры | Ctrl + F11 |
| Переход к закладке (удаление — клавишей Delete) | Shift + F11 |
| Быстрый переход к закладке с присвоенным числом | Ctrl + Число |

***Редактирование:***

|  |  |
| --- | --- |
| Отменить последнее действие | Ctrl + Z |
| Отменить последнюю отмену действия | Ctrl + Shift + Z |
| Расширенная вставка из буфера обмена (с историей) | Ctrl + Shift + V |
| Инкрементальное выделение выражения | Ctrl (+ Shift) + W |
| Перемещение между словами | Ctrl + влево/вправо |
| Прокрутка кода без изменения позиции курсора | Ctrl + вверх/вниз |
| Переход в начало/конец файла | Ctrl + Home/End |
| Удаление строки, отличие в том, где потом окажется курсор | Shift + Del (Ctrl + Y) |
| Удалить от текущей позиции до конца слова | Ctrl + Del |
| Удалить от текущей позиции до начала слова | Ctrl + Backspace |
| Дублировать текущую строку | Ctrl + D |
| Увеличить / уменьшить текущий отступ | Tab / Shift + Tab |
| Выравнивание отступов в коде | Ctrl + Alt + I |
| Приведение кода в соответствие Code-Style | Ctrl + Alt + L |
| Закомментировать/раскомментировать текущую строку | Ctrl + / |
| Закомментировать/раскомментировать выделенный код | Ctrl + Shift + / |
| Фолдинг, свернуть/развернуть | Ctrl + -/+ |
| Фолдинг, свернуть/развернуть все | Ctrl + Shift + -/+ |
| Сделать текущий скоуп сворачиваемым и свернуть его | Ctrl + Shift + . |
| Сделать текущий скоуп несворачиваемым | Ctrl + . |
| Замена в тексте | Ctrl + R |
| Замена во всех файлах | Ctrl + Shift + R |

***Работа с окнами, вкладками:***

|  |  |
| --- | --- |
| Перемещение между вкладками | Alt + влево/вправо |
| Закрыть вкладку | Ctrl + F4 |
| Открытие/закрытие окон Project, Structure, Changes и тд | Altl + число |
| Переключение между вкладками и окнами | Ctrl + Tab |
| Закрыть активное окно | Shift + Esс |
| Открыть последнее закрытое окно | F12 |
| Zoom, если он был вами настроен | Ctrl + колесико мыши |

***Работа с поиском:***

|  |  |
| --- | --- |
| Быстрый поиск по всему проекту | Дважды Shift |
| Быстрый поиск по настройкам, действиям и тд | Ctr + Shift + A |
| Перейти к следующему/предыдущему методу | Alt + вниз/вверх |
| Перемещение к началу и концу текущего скоупа | Ctrl + [ и Ctrl + ] |
| Поиск в файле | Ctrl + F |
| Поиск по всем файлам (переход — F4) | Ctr + Shift + F |
| Искать слово под курсором | Ctrl + F3 |
| Искать вперед/назад | F3 / Shift + F3 |
| Переход к строке или строке:номеру\_символа | Ctrl + G |
| Список методов с переходом к их объявлению | Ctrl + F12 |
| Список недавно открытых файлов с переходом к ним | Ctrl + E |
| Список недавно измененных файлов с переходом к ним | Ctrl + Shift + E |
| Иерархия наследования текущего класса и переход по ней | Ctrl + H |
| Иерархия вызовов выбранного метода | Ctrl + Alt + H |
| Поиска класса по имени и переход к нему | Ctrl + N |
| Поиск файла по имени и переход к нему | Ctrl + Shift + N |
| Перейти к объявлению переменной, класса, метода | Ctrl + B |
| Перейти к реализации | Ctrl + Alt + B |
| Определить тип и перейти к его реализации | Ctrl + Shift + B |
| Перемещение назад по стеку поиска | Shift + Alt + влево |
| Перемещение вперед по стеку поиска | Shift + Alt + вправо |
| Переход к следующей / предыдущей ошибке | F2 / Shift + F2 |
| Найти все места, где используется метод / переменная | Shift + Alt + 7 |
| Как предыдущий пункт, только во всплывающем окне | Ctrl + Alt + 7 |

***Генерация кода и рефакторинг:***

|  |  |
| --- | --- |
| Полный автокомплит | Ctrl + Space |
| Автокомплит с фильтрацией по подходящему типу | Ctrl + Shift + Space |
| Простой автокомплит по словам, встречающимся в проекте | Alt + / |
| Реализовать интерфейс | Ctrl + I |
| Переопределить метод родительского класса | Ctrl + O |
| Генерация шаблонного кода (обход по итератору и тд) | Ctrl + J |
| Обернуть выделенный код в один из шаблонов | Ctrl + Alt + J |
| Генератор кода — сеттеров, зависимостей в pom.xml и тд | Alt + Insert |
| Переименование переменной, класса и тд во всем коде | Shift + F6 |
| Изменение сигнатуры метода во всем коде | Ctrl + F6 |
| Перемещение метода, класса или пакета | F6 |
| Создать копию класса, файла или каталога | F5 |
| Создать копию класса в том же пакете | Shift + F5 |
| Безопасное удаление класса, метода или атрибута | Alt + Delete |
| Выделение метода | Ctrl + Alt + M |
| Выделение переменной | Ctrl + Alt + V |
| Выделение атрибута | Ctrl + Alt + F |
| Выделение константы (public final static) | Ctrl + Alt + C |
| Выделение аргумента метода | Ctrl + Alt + P |
| Инлайнинг метода, переменной, аргумента или константы | Ctrl + Alt + N |
| Оптимизация импортов | Ctrl + Alt + O |

**Система контроля версий**

Активировать работу с системой контроля версий в проекте можно через вкладку «VCS», Pycharm умеет работать с Git, Mercurial, CVS, Perforce, Subversion.

После выбора, к примеру, Git, Pycharm создает папку .git в корне проекта.

После этого IDE начинает подсвечивать красным неверсионированные файлы, через тоже самое меню VCS можно сделать commit, push, pull итд.

Также можно склонировать уже существующий проект и работать с ним в Pycharm, для этого при создании нового проекта надо указать, что достать его надо из репозитория. При необходимости можно сравнить файлы, лежащие в git с версией на машине, для разрешения каких-либо конфликтов и т. д.

**Стилизация**

**Code style**

В pycharme есть возможность самому определять внешний вид кода, настраивая размер табуляции, фон и т.д. Для этого в настройках надо выбрать пункт Code style.

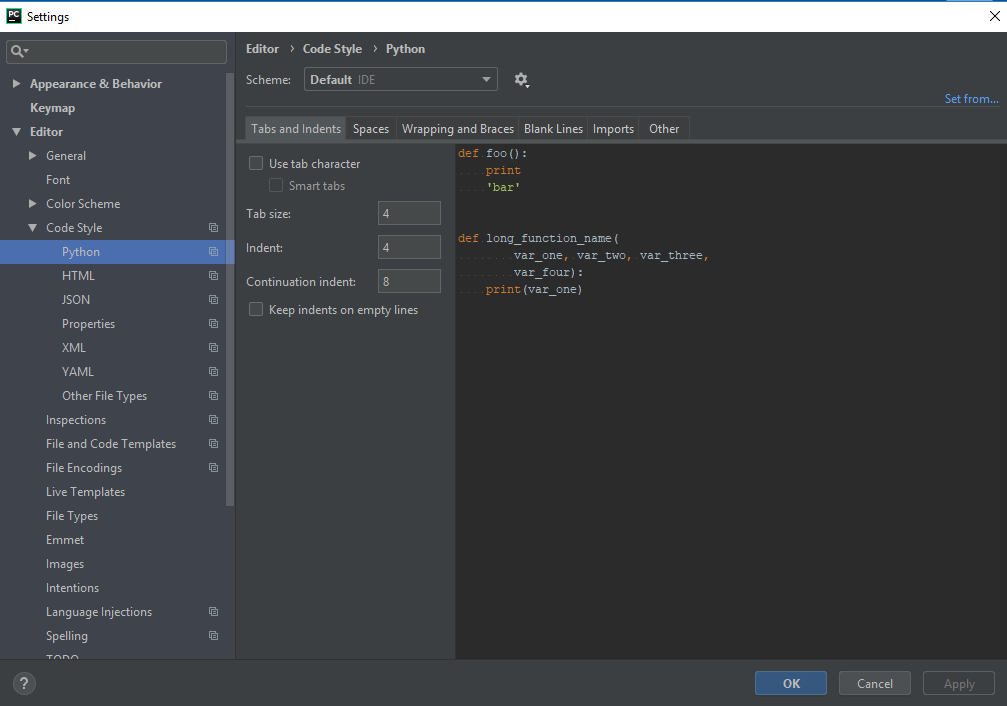
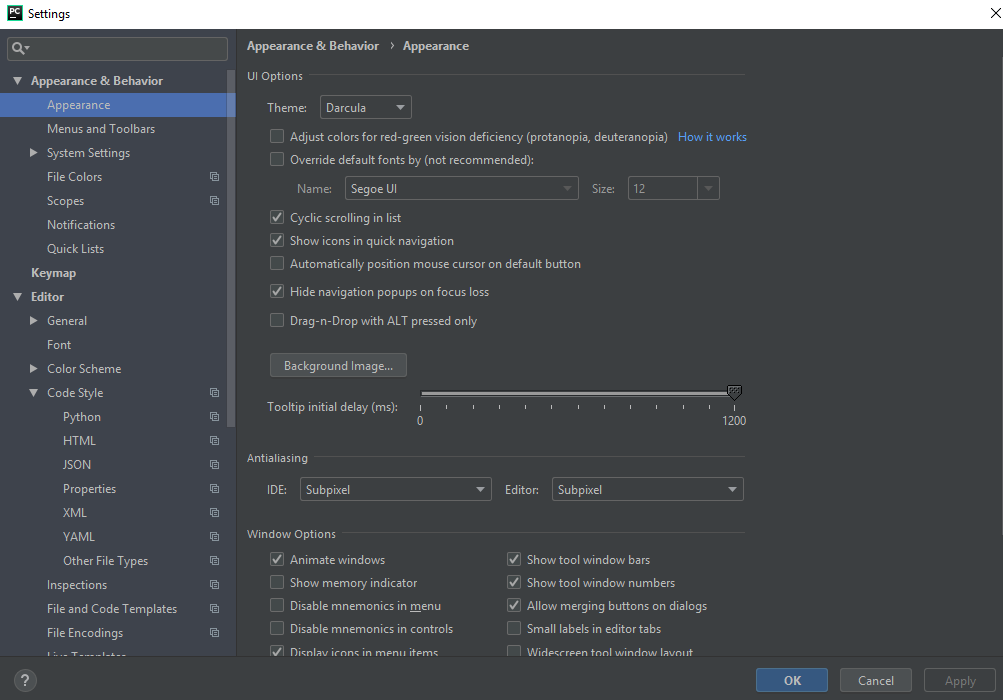


Рис. 6 Настройка табуляции кода

**IDE style**

Также ещё при первом открытие IDE предлагается выбрать фон, в котором будем работать. Более подробные настройки внешнего вида IDE можно найти на вкладке Appearance в настройках. Тут можно также добавить некоторые пункты в меню, поменять цвет темы и другое.



**Вывод**

Pycharm является одной из лучшей сред разработки для разработчика. Она даёт широкий простор в настройке поведения IDE под себя, что позволяет создать комфортные условия для написания кода. В то же время IDE помогает разработчику, давая советы по улучшению и оптимизации кода.