

**Министр науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования «Национальный
исследовательский университет ИТМО»**

Факультет информационных технологий и программирования

Цифровая культура

Лабораторная работа №1

Выполнил студент группы № М3111

Акберов Рустам Ханкишиевич

Подпись:

A handwritten signature in black ink, consisting of a stylized, cursive letter 'A' followed by a series of loops and a long horizontal stroke extending to the right.

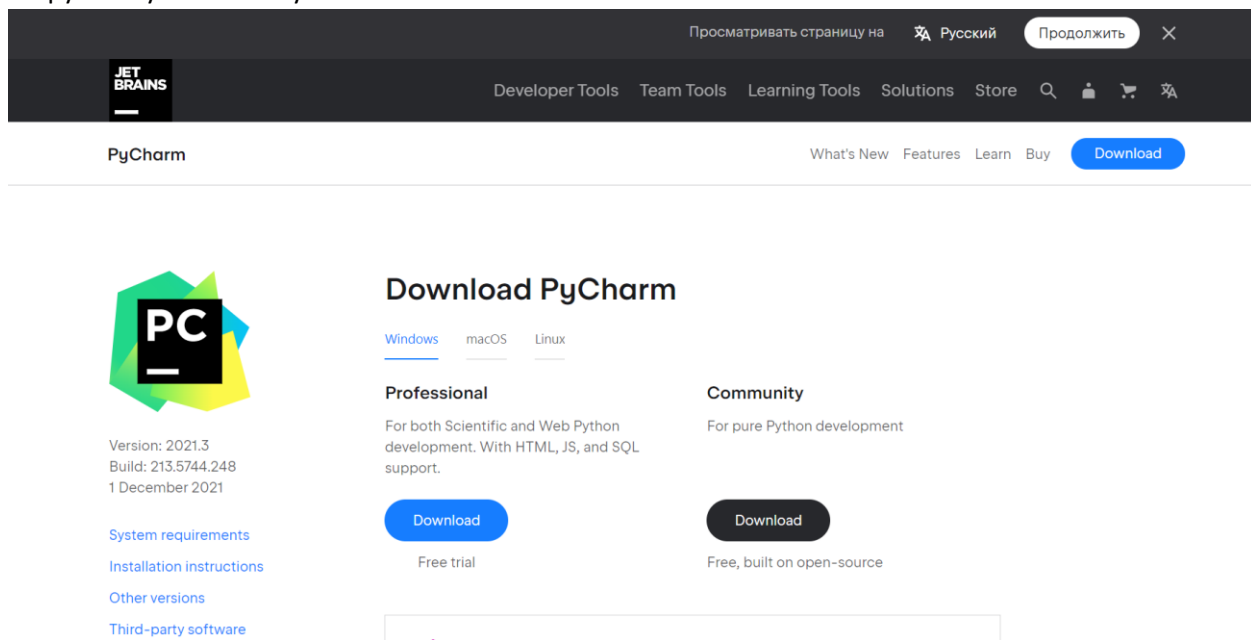
Проверил:

Санкт-Петербург

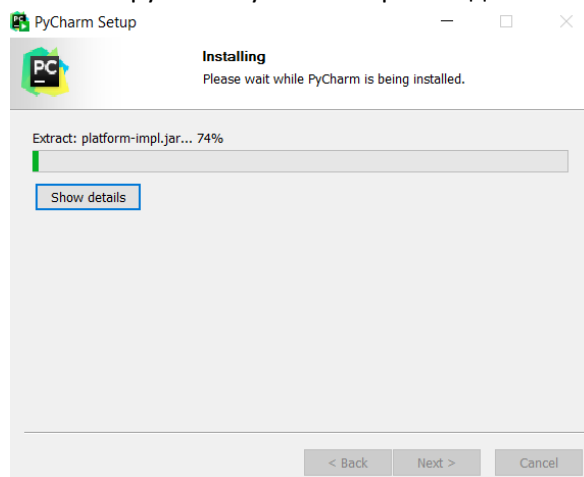
2021

IDE – PyCharm (JetBrains)

1) Загрузка и установка PyCharm



На официальном сайте JetBrains скачиваем PyCharm (professional по подписке)
После загрузки запускаем и производим быструю установку

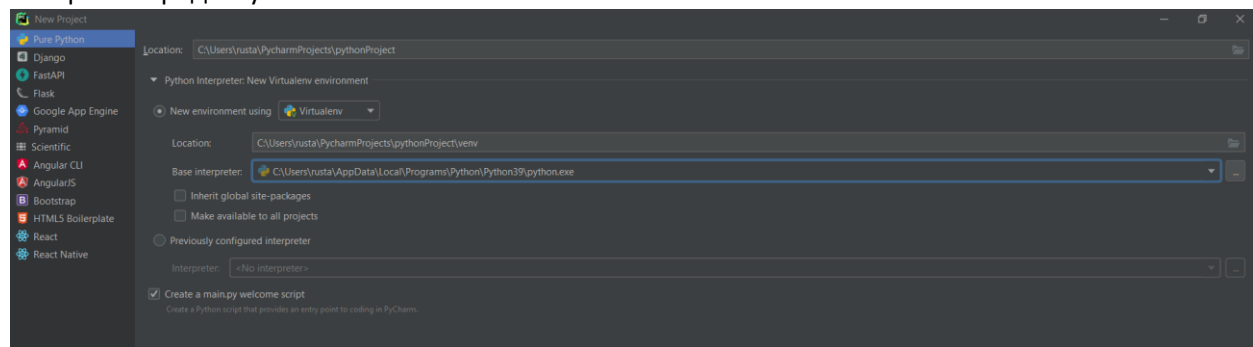


2) Одновременно с этим качаем интерпретатор python с официального сайта и устанавливаем его





3) Настройка среды PyCharm



В настройках проекта нужно указать путь к интерпретатору (в нашем случае python, заранее установленный с официального сайта www.python.org)

4) Калькулятор

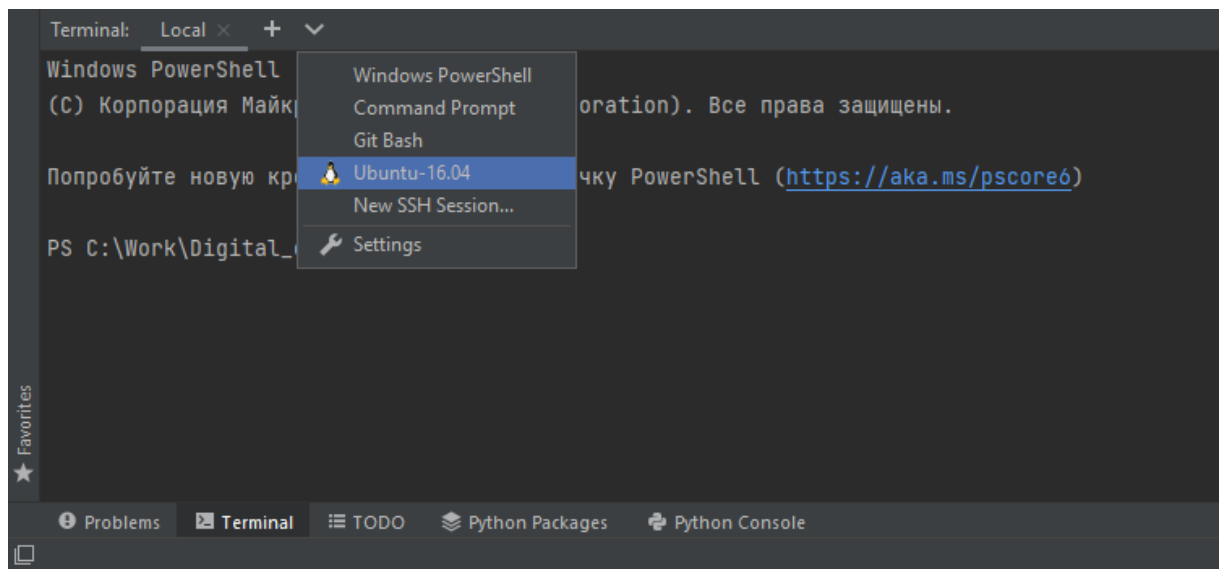
```
1 print("Ноль в качестве знака операции"
2      "\nзавершит работу программы")
3 while True:
4     s = input("Знак (+, -, *, /): ")
5     if s == '0':
6         break
7     if s in ('+', '-', '*', '/'):
8         x = float(input("x="))
9         y = float(input("y="))
10        if s == '+':
11            print("%.2f" % (x+y))
12        elif s == '-':
13            print("%.2f" % (x-y))
14        elif s == '*':
15            print("%.2f" % (x*y))
16        elif s == '/':
17            if y != 0:
18                print("%.2f" % (x/y))
19            else:
20                print("Деление на ноль!")
21        else:
22            print("Неверный знак операции!")
```

5) Тесты к калькулятору

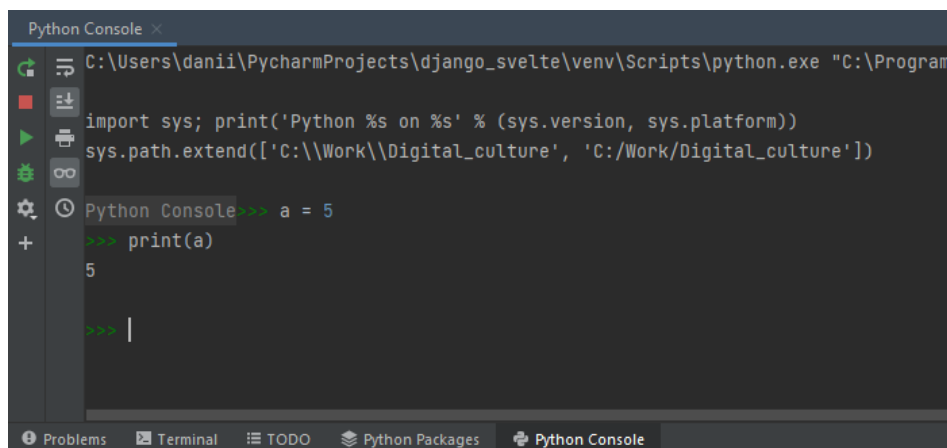
```
C:\Users\rusta\PycharmProjects\p
Ноль в качестве знака операции
завершит работу программы
Знак (+,-,*,/): +
x= 17
y= 17
17.00
Знак (+,-,*,/): /
x= 17
y= 17
1.12
Знак (+,-,*,/): *
x= 17
y= 17
80.00
Знак (+,-,*,/): |
```

6) Терминал

Существует 5 типов терминала: Windows PowerShell(PowerShell), command prompt (cmd), Git Bash (git session and commands), Ubuntu, SSH Session.



Также существует Python Console, где можно писать построчный код и исполнять его.



Сниппеты:

```
"""Добавление точки в конец строки"""

def ending(line):
    if line[-1] != '.':
        line += '.'
    return line

"""Строка с заглавной буквы"""

def start_line(line):
    return line[0].upper() + line[1:]

"""Смена значений"""

def swap(a, b):
    return b, a

"""Из строки чисел в массив чисел"""

def str_numbers(line):
    return [int(s) for s in list(line)]

"""Состав букв в слове"""

def about_str(line):
    response = {}
    for el in line:
        if el in response:
            response[el] += 1
        else:
            response[el] = 1
    return response
```

7) Наиболее распространённые HotKeys

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

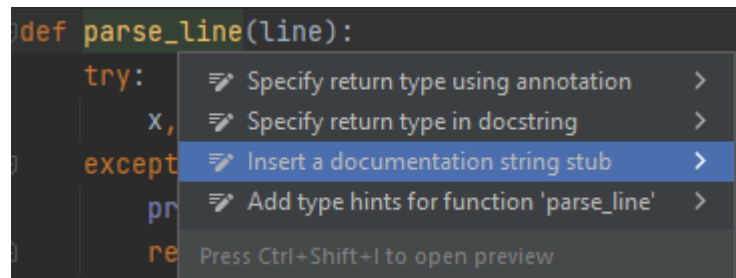
8) Задокументировать код с помощью расширения

```
def calc(x, operator, y):
    """
    :param x:
    :type x:
    :param operator:
    :type operator:
    :param y:
    :type y:
    :return:
    :rtype:
    """
    global prev
    if operator == '+':
        print(x + y)
        prev = x + y
    elif operator == '-':
        print(x - y)
        prev = x - y
    elif operator == '*':
        print(x * y)
        prev = x * y
    elif operator == '/':
        print(x / y)
        prev = x / y
    elif operator == '^':
        print(x ^ y)
        prev = x ^ y
    elif operator == '**':
        print(x ** y)
        prev = x ** y
    elif operator == '//':
        print(x // y)
        prev = x // y
    elif operator == '%':
        print(x % y)
        prev = x % y
    else:
        print('Unknow operator')
        prev = None
    return prev

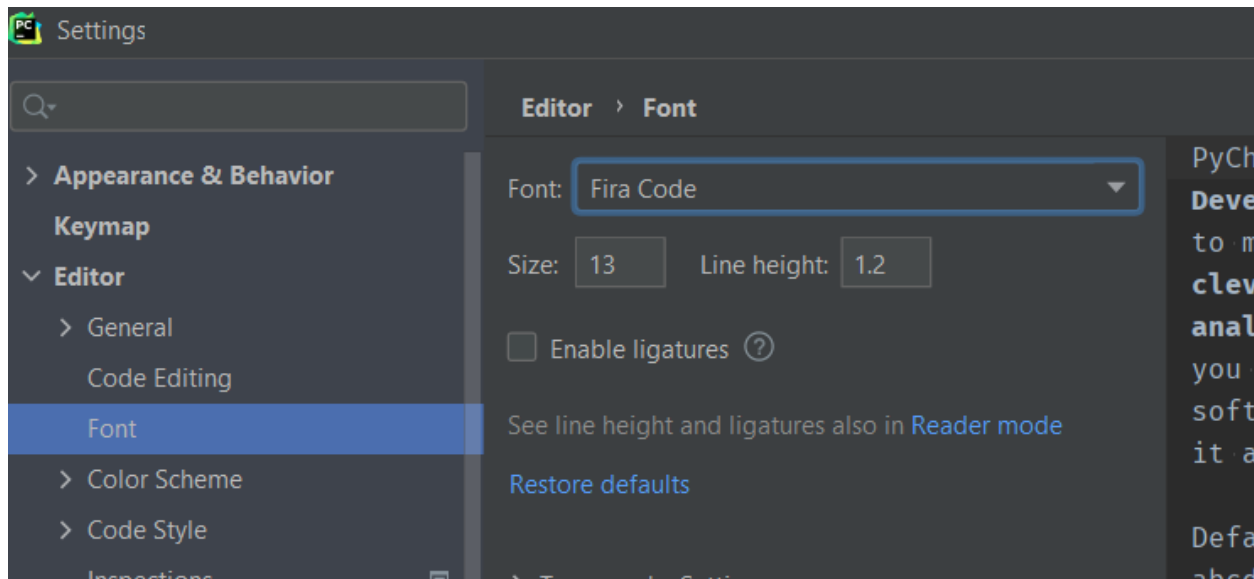
def parse_line(line):
    """
    :param line:
    :type line:
    :return:
    :rtype:
    """
    try:
        x, operator, y = line.split()
    except ValueError as e:
        print('Enter values with whitespaces')
        return
    try:
        x = int(x)
    except ValueError as e:
        x = prev
    y = int(y)
    return [x, operator, y]
```

```
def main(line):  
    """  
  
    :param line:  
    :type line:  
    :return:  
    :rtype:  
    """  
    parsed = parse_line(line)  
    if parsed:  
        return calc(parsed[0], parsed[1], parsed[2])  
    return None
```

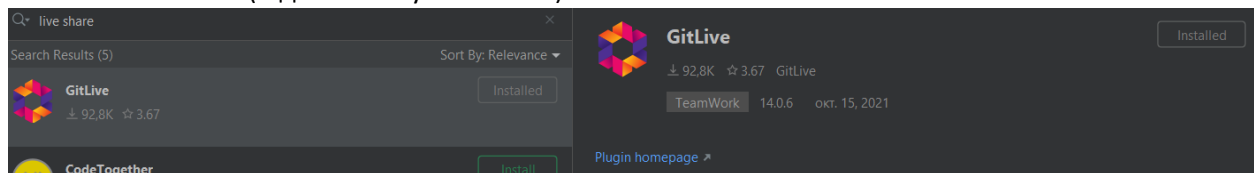
Пример как сделать документирование кода в PyCharm



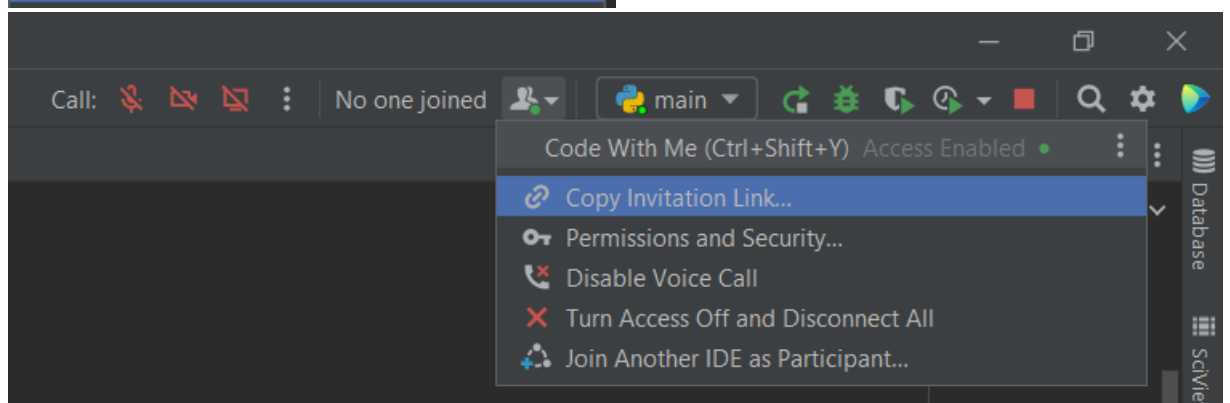
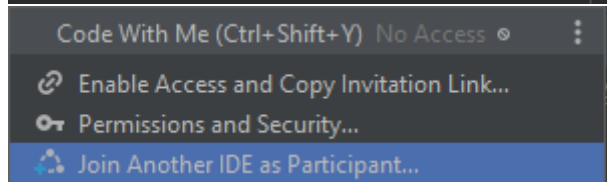
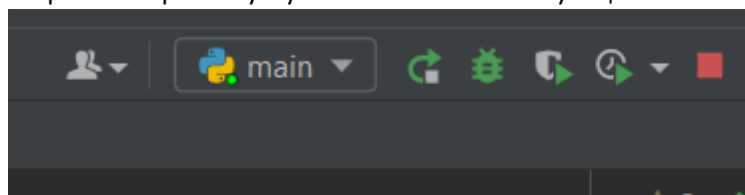
9) Установить шрифт FiraCode



10) Установить плагин (В данном случае GitLive)



В правом верхнем углу появится соответствующий значок



Данный плагин позволяет осуществлять совместную разработку проекта, созваниваться, редактировать код будучи в разных файлах, однако, 'гость' не может создавать или удалять файлы создателя.