

Федеральное государственное автономное
образовательное учреждение высшего
образования
«Национальный исследовательский университет
ИТМО»

Факультет Информационных технологий и программирования

Работа: Игнатов М. В., М3112, Инструментальные средства
разработки ПО, работа №5

Выполнил: Игнатов Максим Вячеславович
Проверил: Повышев Владислав Вячеславович

Санкт-Петербург
2022 г.

1. Скачиваем PyCharm от JetBrains на официальном сайте (<https://www.jetbrains.com/ru-ru/pycharm/download/#section=mac>)



Версия: 2022.2.3
Сборка: 222.4345.23
11 октября 2022 г.

[Системные требования](#)
[Инструкция по установке](#)
[Другие версии](#)
[Стороннее ПО](#)

Скачать PyCharm

Windows **macOS** Linux

Professional

Для научной и веб-разработки на Python. Поддерживает HTML, JS и SQL.

Скачать

.dmg (Intel)

Доступна бесплатная пробная версия на 30 дней

Community

Для разработки только на Python

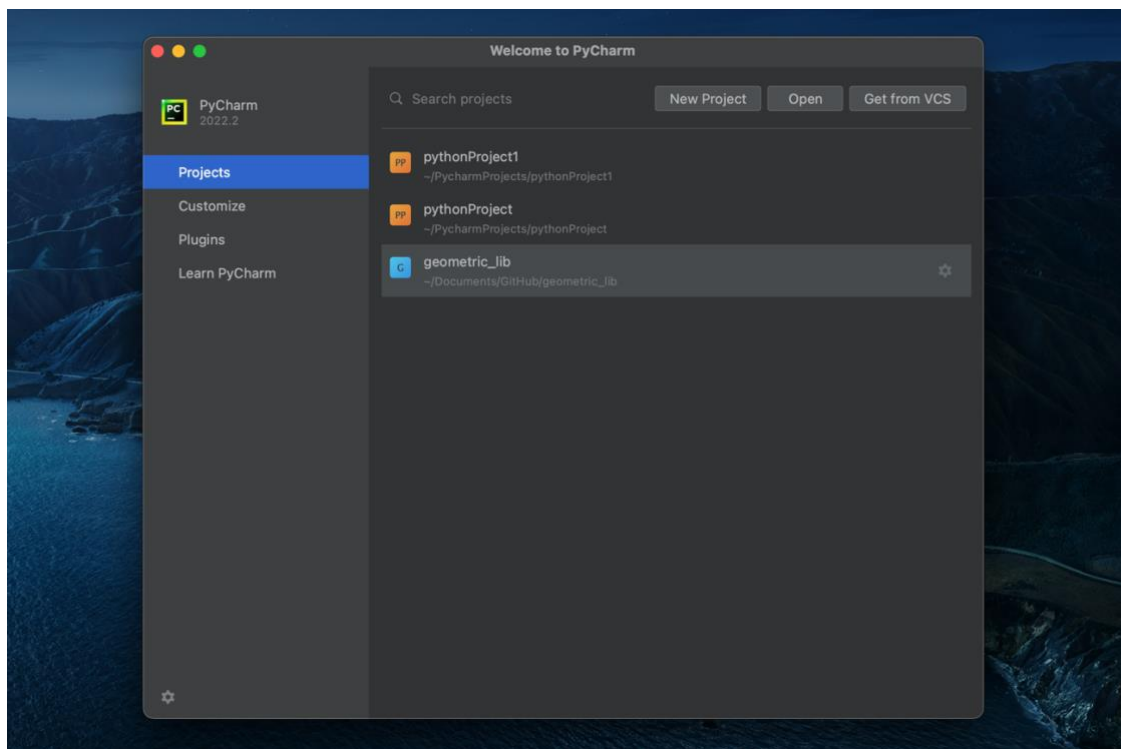
Скачать

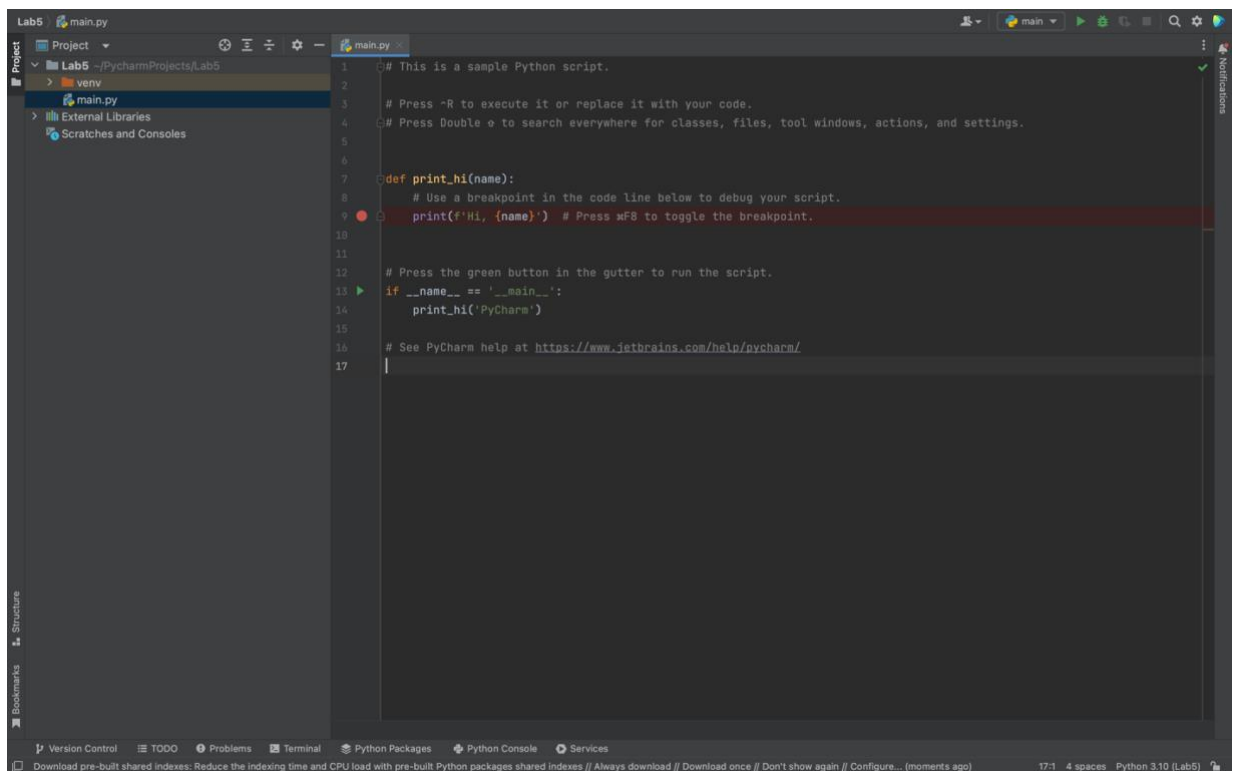
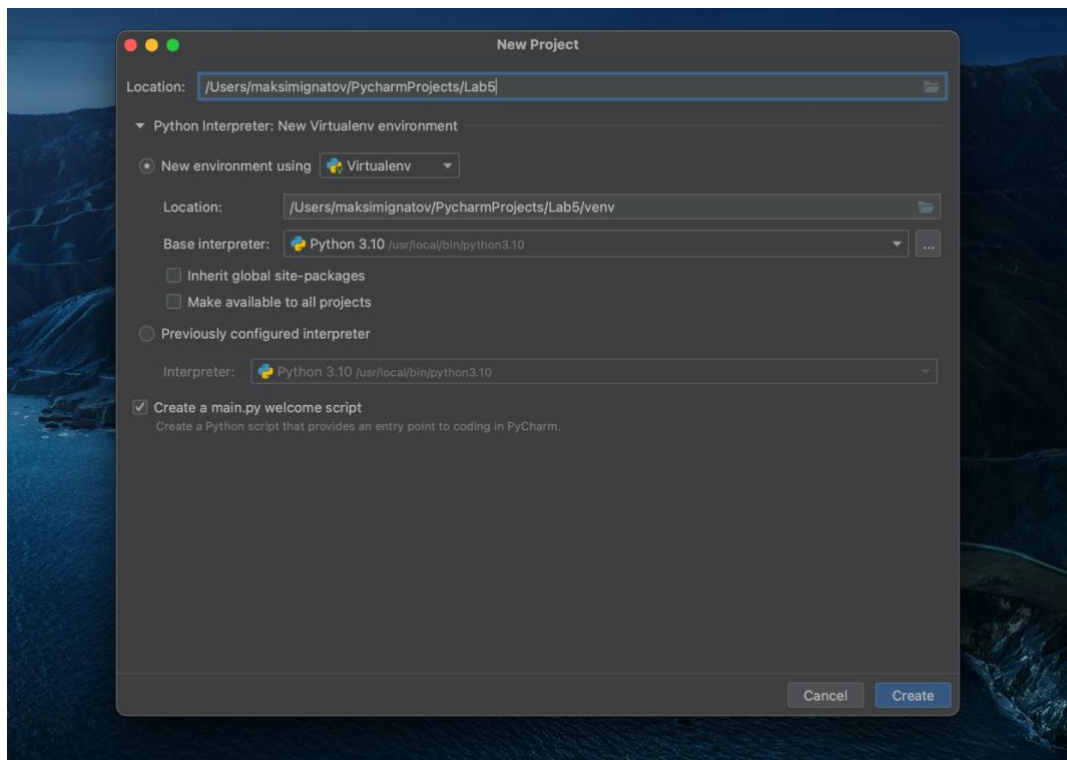
.dmg (Intel)

Бесплатная, с открытым кодом

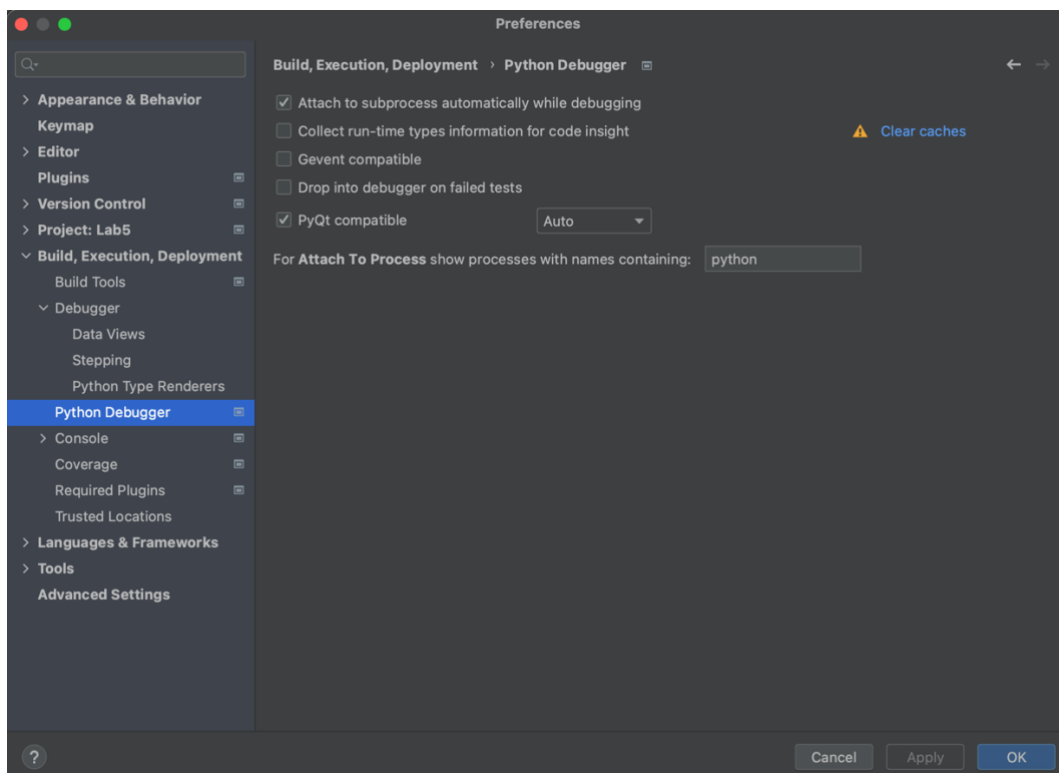
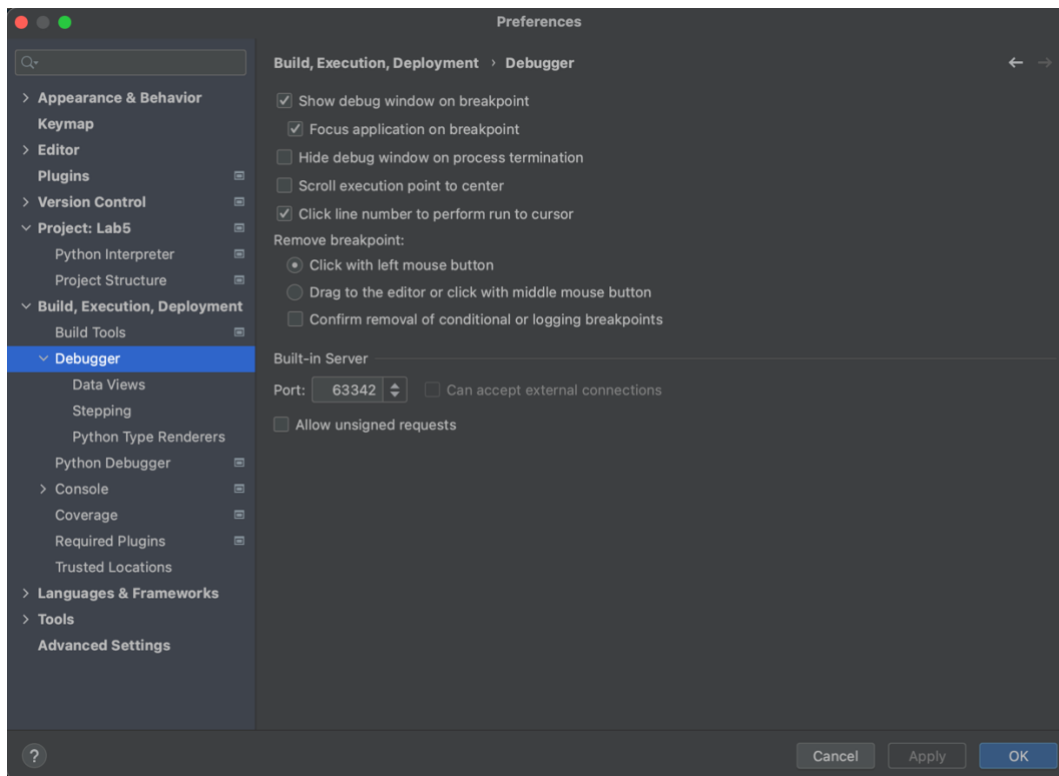
Выберите установщик для Intel или Apple Silicon

2. Заходим в IDE и создаем новый проект





3. Конфигурация для запуска и отладки кода в PyCharm происходит автоматически



4. Напишем простой калькулятор чтобы на его примере продемонстрировать работу юнит тестов

```
1 class Calculator:
2
3     def __init__(self):
4         pass
5
6     def sum(self, a, b):
7         result = a + b
8         return result
9
10
11     def subtract(self, a, b):
12         result = a - b
13         return result
14
15
16     def multiply(self, a, b):
17         if b > 10000000:
18             return 'Слишком большая степень'
19         else:
20             result = a * b
21             return result
22
23
24     def divide(self, a, b):
25         if b != 0:
26             result = a / b
27             return result
28         else:
29             return 'Деление на 0'
30
31
32 def calculate(a, b, operation, self=None):
33     result = None
34
35     if operation == '+':
36         result = Calculator.sum(self, a, b)
37     elif operation == '-':
38         result = Calculator.subtract(self, a, b)
39     elif operation == '/':
40         result = Calculator.divide(self, a, b)
41     elif operation == '*':
```

```

37         result = Calculator.divide(self, a, b)
38     elif operation == '*':
39         result = Calculator.multiply(self, a, b)
40     else:
41         print('Неизвестная операция')
42
43     return result
44
45
46 def ask_operation():
47     message = ''
48     Пожалуйста, введите символ операции, которую вы хотите совершить и нажмите Enter:
49
50     + : Сложение
51     - : Вычитание
52     / : Деление
53     * : Умножение
54
55     Ваш выбор:
56     ''
57
58     operation = input(message)
59
60     return operation
61
62
63 def run_calculator():
64     a = int(input('Введите первое число: '))
65     b = int(input('Введите второе число: '))
66
67     operation = ask_operation()
68
69     result = calculate(a, b, operation)
70
71     print(f'Результат вычислений: {result}')
72
73     run_calculator()

```

- Для написания тестов воспользуемся пакетом unittest. Для каждой из функций, находящихся в программе, зададим собственные входные данные (с помощью self) и проверим условие с помощью встроенного метода assertEquals. Необходимо также подписать перед каждой из функций «test», чтобы показать питону, что это метод тестирования.

```

1  import unittest
2
3  from Calculator import Calculator
4
5
6  class testCalculator(unittest.TestCase):
7
8      def setUp(self):
9          self.calculator = Calculator()
10
11     def test_sum(self):
12         self.assertEqual(self.calculator.sum(4, 2), 6)
13
14     def test_subtract(self):
15         self.assertEqual(self.calculator.subtract(10, 8), 2)
16
17     def test_multiply(self):
18         self.assertEqual(self.calculator.multiply(7, 5), 35)
19
20     def test_divide(self):
21         self.assertEqual(self.calculator.divide(10, 2), 5)
22
23
24     if __name__ == "__main__":
25         unittest.main()
26

```

Список горячих клавиш на Python

Ctrl + D — дублирует строку

Ctrl + Shift + U — делает выделенный текст или заглавными, или строчными буквами

Ctrl + Delete — удаляет строку, в конце которой находится курсор

Ctrl + Alt + L — форматирование кода соответствии с настройками code style.

Ctrl + Shift + V — вставка из истории буфера

Ctrl + / — закомментировать/раскомментировать текущую строку.

Ctrl + [или] — перемещает курсор в начало или конец блока с кодом, в теле функции, в теле цикла и т.д.

Tab — сделать отступ (табуляцию)

SHIFT + Tab — убрать отступ

Ctrl + Shift + Стрелка вверх — перемещает текущий логический блок вверх

Ctrl + Shift + Стрелка вниз — перемещает текущий логический блок вниз

Ctrl + курсор мыши над кодом — краткая информация

Двойной Shift — поиск по проекту

Ctrl + F — поиск в текущем файле

Ctrl + R — поиск и замена

Ctrl+Shift+E — вызовет всплывающее окно, в котором показаны последние куски кода в разных файлах, с которыми вы работали

Ctrl + Shift + F — поиск в группе файлов (например в проекте или папке)

Ctrl + Shift + R — поиск и замена в группе файлов

Ctrl + Shift + N — поиск файла по названию