

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №16
дисциплины «Основы программной инженерии»

Выполнил:
Звездин Алексей Сергеевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Ход работы

1. Я изучил теоретический материал работы

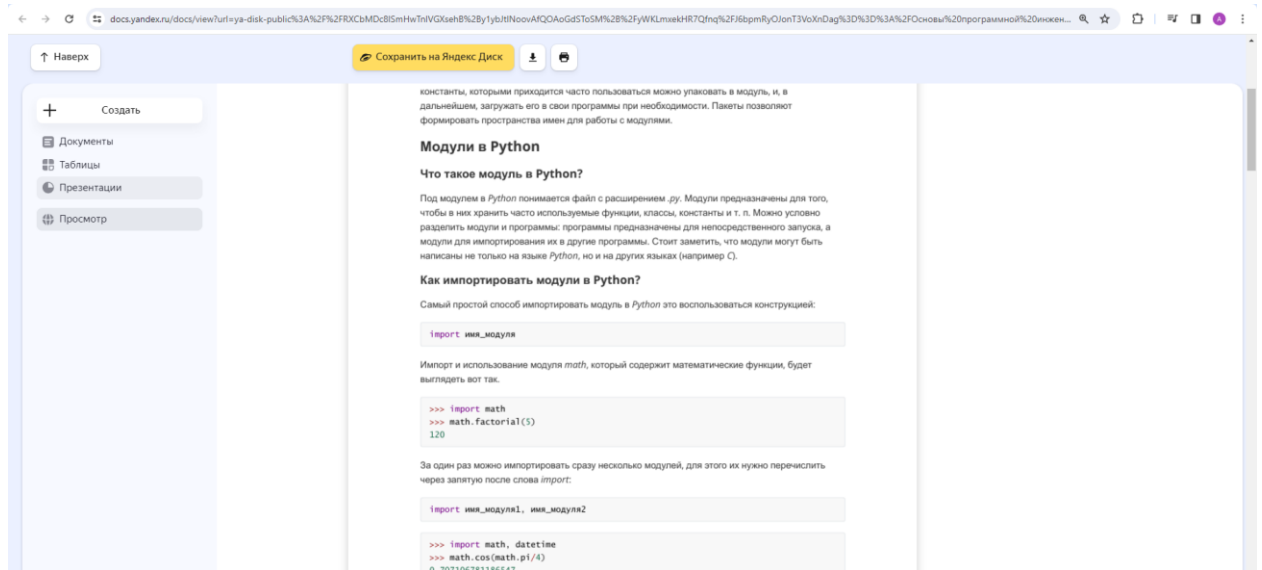


Рисунок 1.1 – Изучение материала для лабораторной работы

2. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python

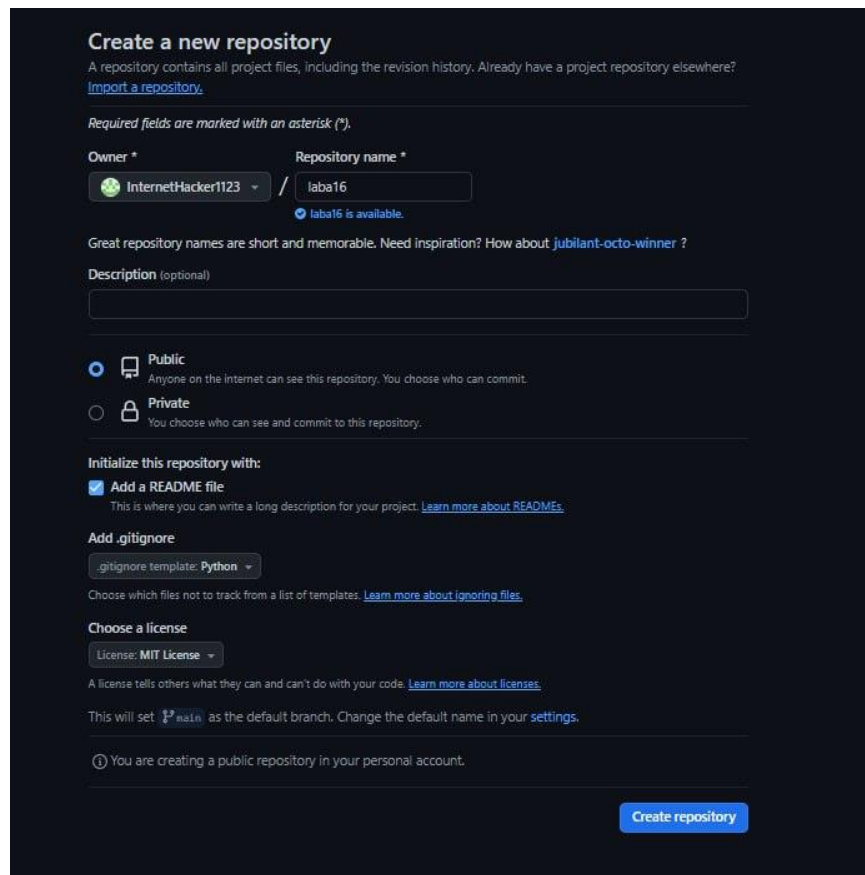


Рисунок 2.1 – Настройка репозитория

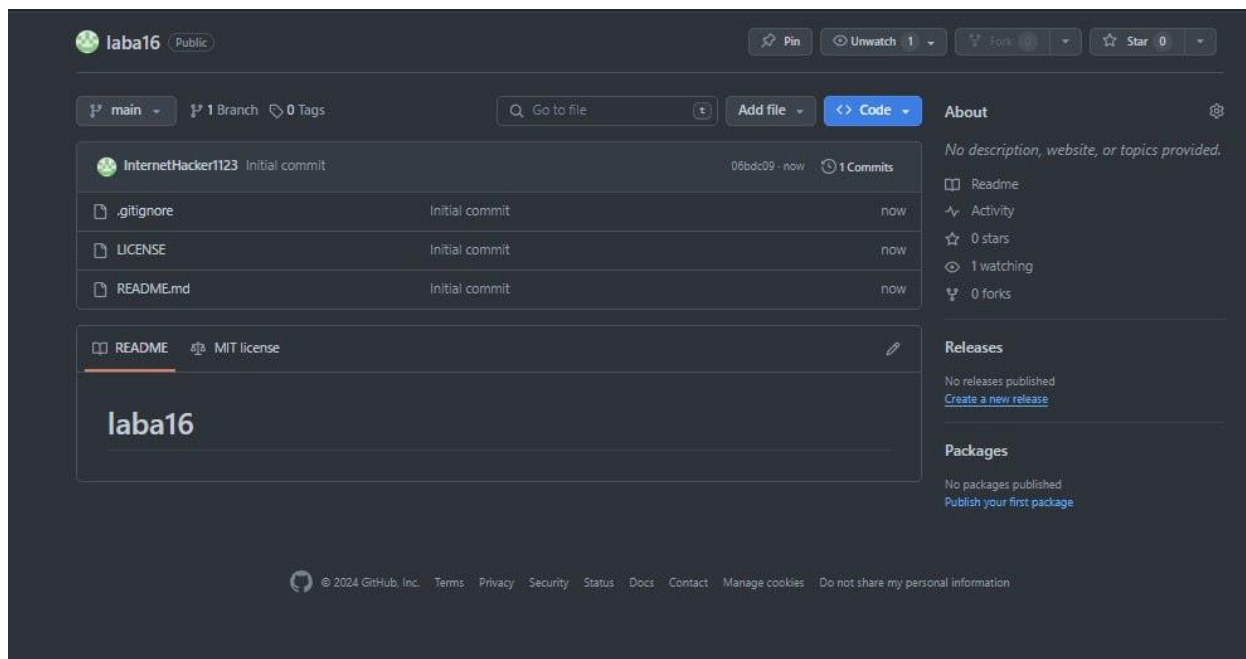


Рисунок 2.2 – Готовый репозиторий

3. Выполняю клонирование созданного репозитория

```
C:\Windows\system32\cmd.exe
C:\Users\dinoz\OneDrive\Рабочий стол\SE\laba16>git clone https://github.com/InternetHacker1123/laba16.git
Cloning into 'laba16'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
C:\Users\dinoz\OneDrive\Рабочий стол\SE\laba16>
```

Рисунок 3.1 – Клонирование репозитория на локальный диск

4. Дополнил файл .gitignore необходимыми правилами для работы с VS Code

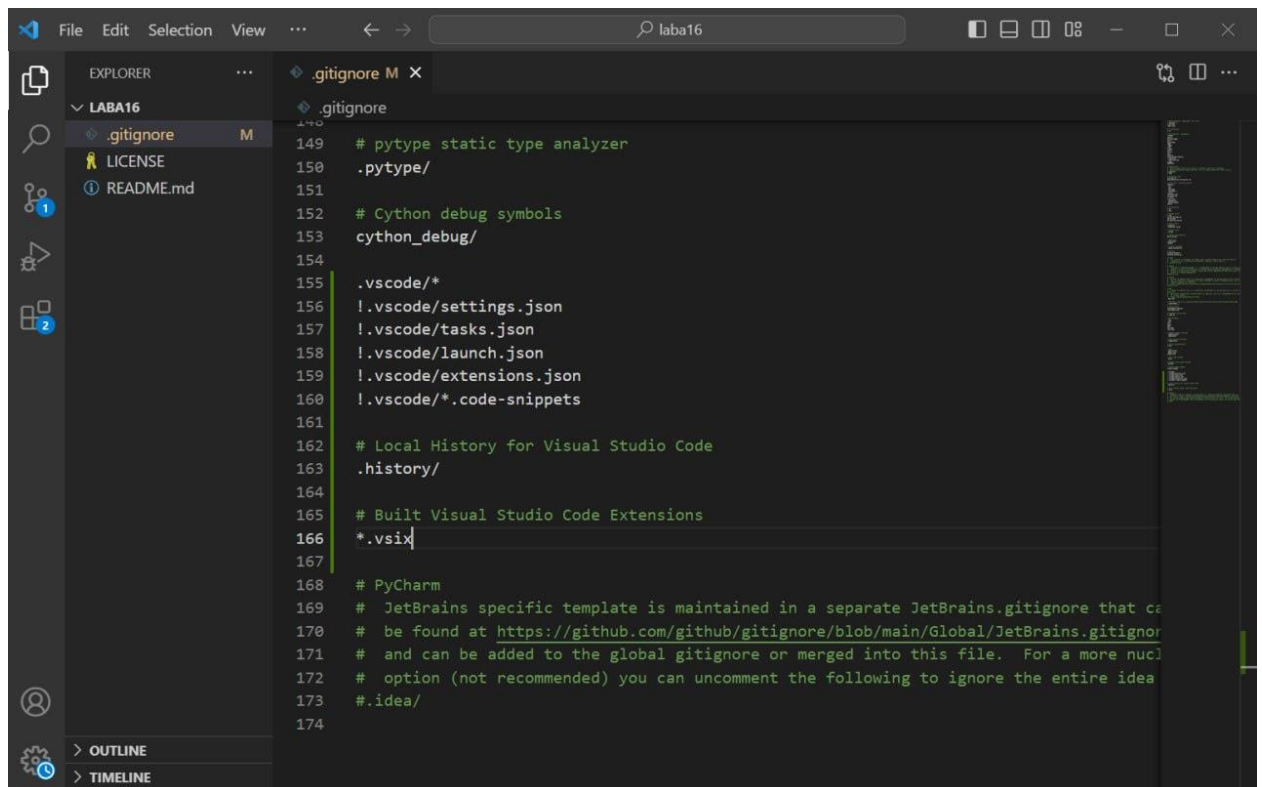


Рисунок 4.1 – .gitignore для VS code

5. Организовал свой репозиторий в соответствии с моделью ветвления git-flow

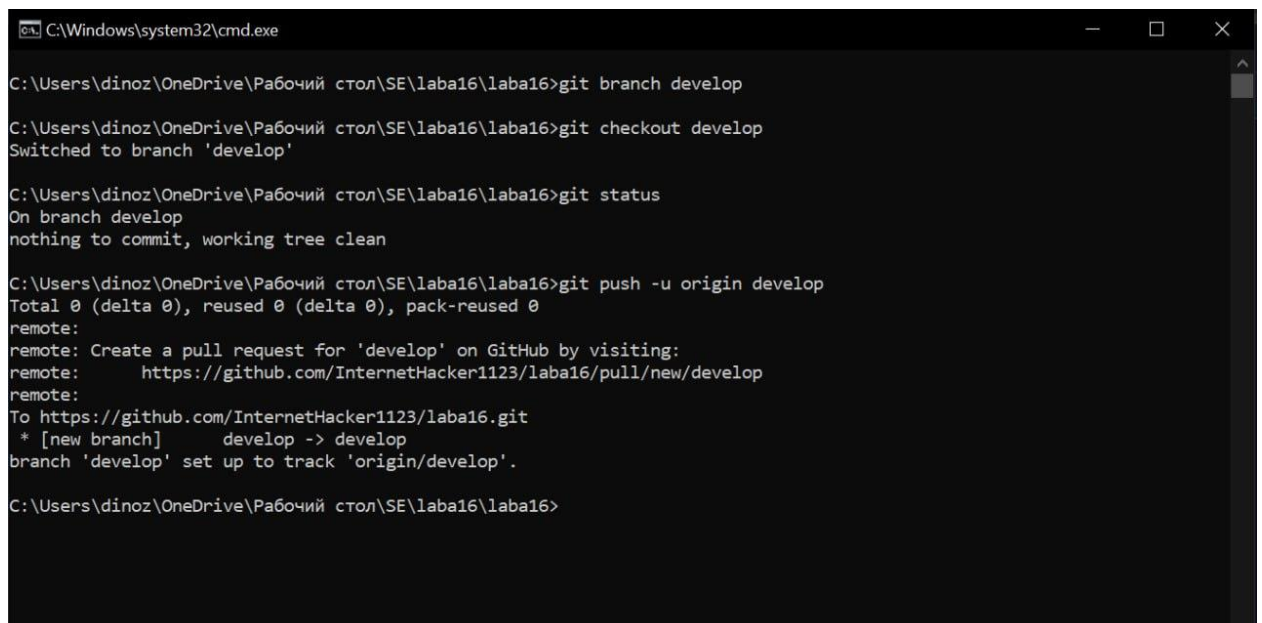


Рисунок 5.1 – Создание ветки develop от ветки main

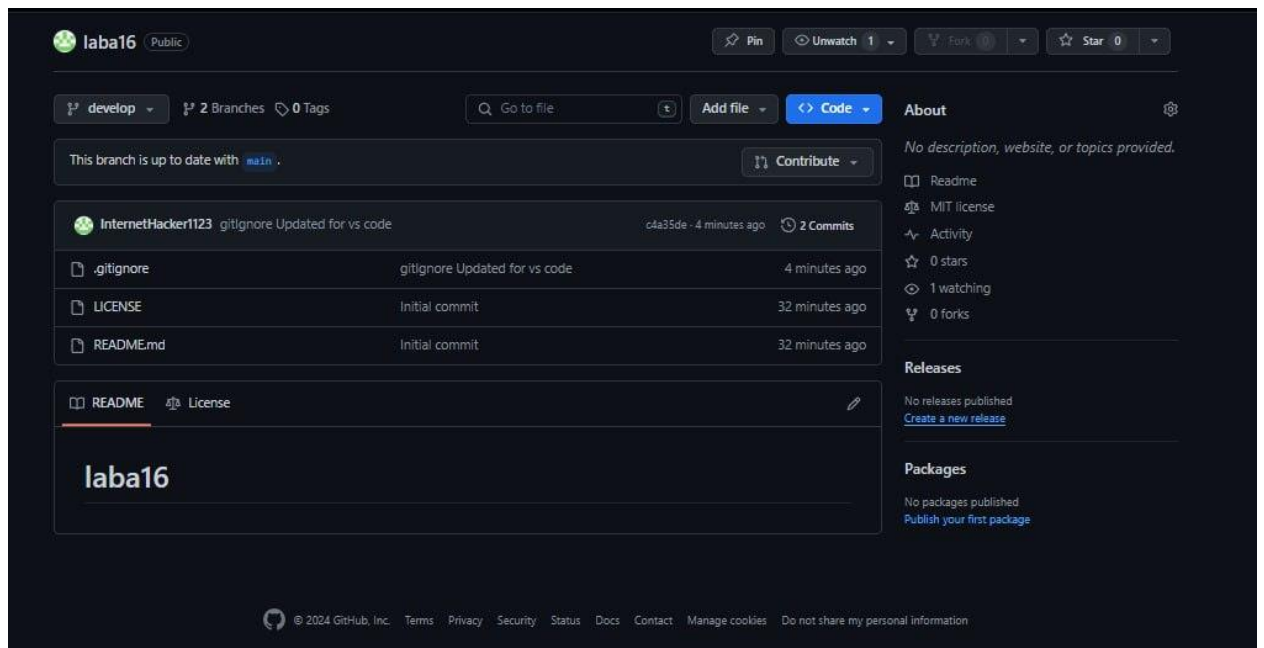


Рисунок 5.2 – Ветка develop на GitHub

6. Создал проект PyCharm в папке репозитория

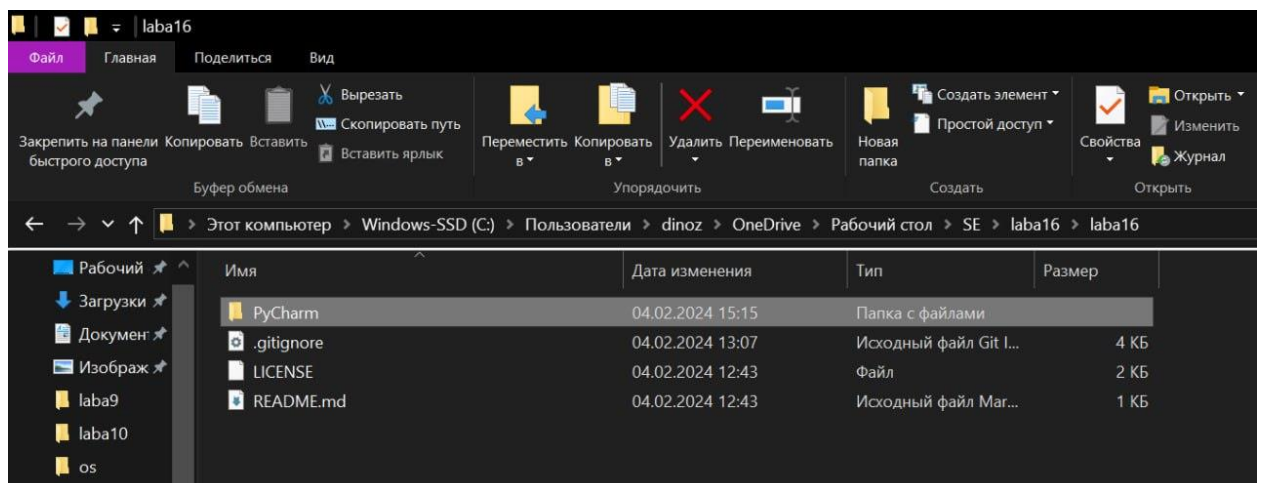


Рисунок 6.1 – Репозиторий с проектом PyCharm

7. Выполнил индивидуальные задания. Привел в отчете скриншоты работы программ решения индивидуального задания.

```

PyCharm > task1 > module.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def outer_function(f):
5      def inner_function(a, b):
6          result = f(a, b)
7          return f"Для значений {a}, {b} функция f({a}, {b}) = {result}"
8      return inner_function
9
10
11 # Пример функции f, которую мы передадим во внешнюю функцию
12 def my_function(a, b):
13     return a * b + a - b
14
15
16 # Главная функция
17 def main():

```

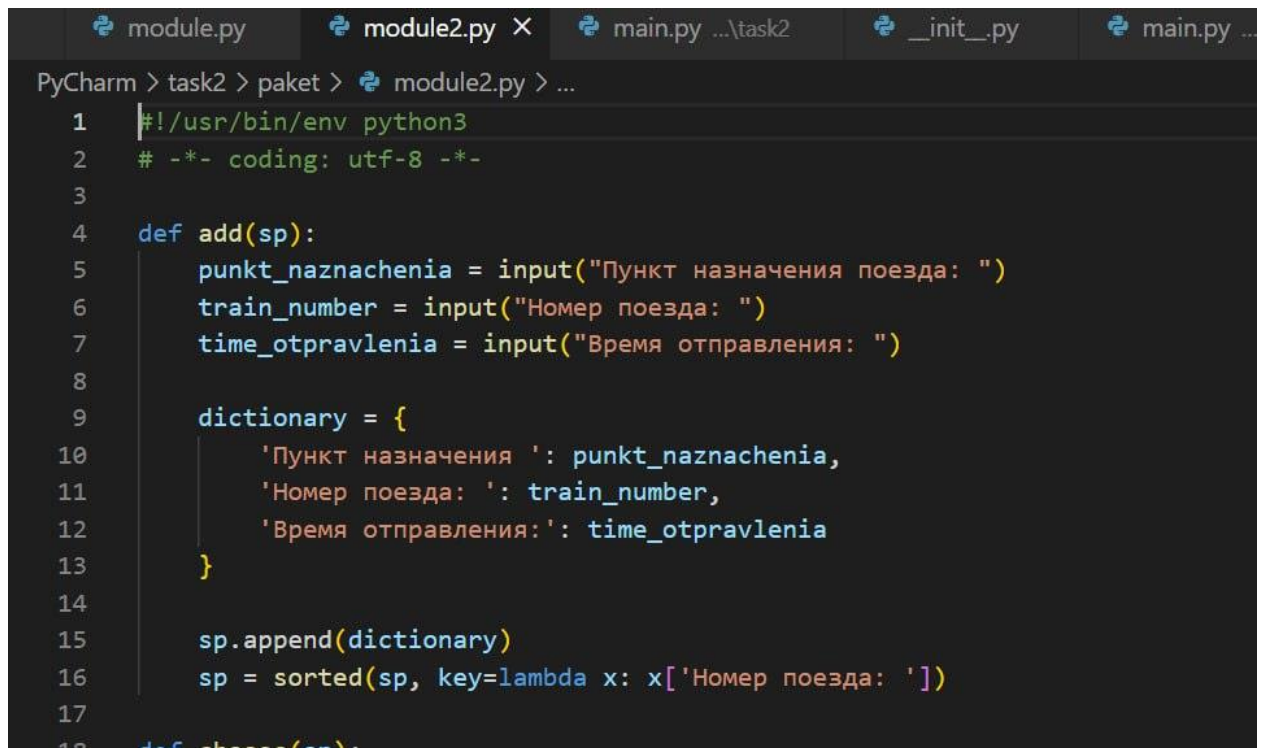
Рисунок 7.1 – Код программы с функциями из лабораторной № 2.11

```

module.py  module2.py U  main.py ...\task2 U
PyCharm > task1 > main.py
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import module
5
6  if __name__ == "__main__":
7      module.main()

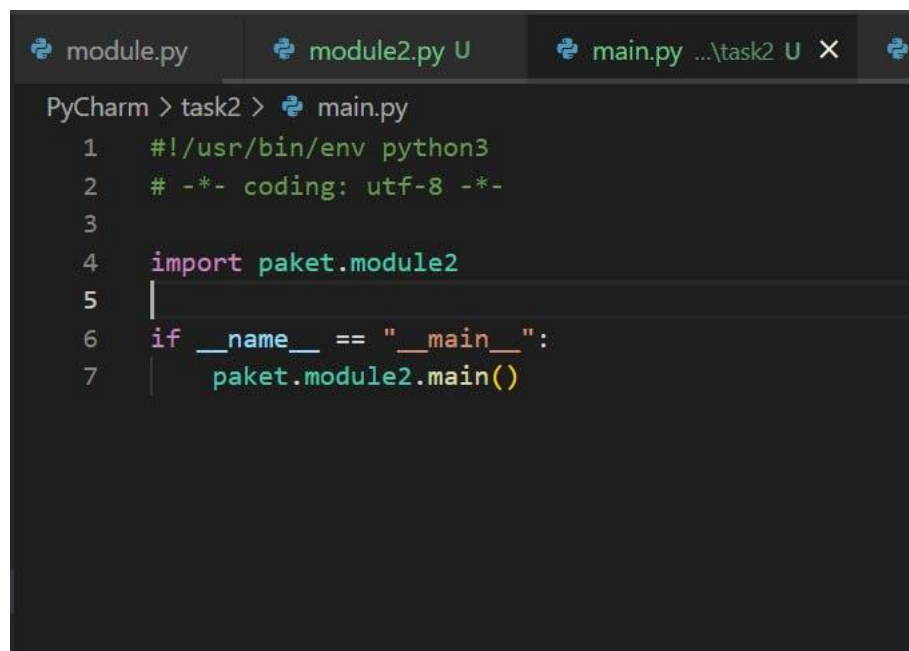
```

Рисунок 7.2 – Основная программа, с импортированием функций из лабораторной № 2.11



```
PyCharm > task2 > paket > module2.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def add(sp):
5      punkt_naznachenia = input("Пункт назначения поезда: ")
6      train_number = input("Номер поезда: ")
7      time_otpravleniya = input("Время отправления: ")
8
9      dictionary = {
10         'Пункт назначения ': punkt_naznachenia,
11         'Номер поезда: ': train_number,
12         'Время отправления: ': time_otpravleniya
13     }
14
15     sp.append(dictionary)
16     sp = sorted(sp, key=lambda x: x['Номер поезда: '])
17
18 def choose(sp):
```

Рисунок 7.3 – Основная программа, с использованием функций пакета из лабораторной № 2.8



```
PyCharm > task2 > main.py
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import paket.module2
5
6  if __name__ == "__main__":
7      paket.module2.main()
```

Рисунок 7.4 – Импортирование всех функций пакета из лабораторной № 2.8


```

PyCharm > task2 > paket > __init__.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  __all__ = ["module2"]

```

Рисунок 7.5 – Переменная `__all__` , хранящая список модулей, который импортируется при загрузке через конструкцию

8. Зафиксировал сделанные изменения в репозитории.

```

C:\Users\dinoz\OneDrive\Рабочий стол\SE\laba16\laba16>git merge develop
Updating c4a35de..5eb3809
Fast-forward
 PyCharm/task1/main.py      | 7 +++
 PyCharm/task1/module.py   | 21 ++++++++
 PyCharm/task2/main.py     | 7 +++
 PyCharm/task2/paket/__init__.py | 4 +++
 PyCharm/task2/paket/module2.py | 75 +++++
 5 files changed, 114 insertions(+)
 create mode 100644 PyCharm/task1/main.py
 create mode 100644 PyCharm/task1/module.py
 create mode 100644 PyCharm/task2/main.py
 create mode 100644 PyCharm/task2/paket/__init__.py
 create mode 100644 PyCharm/task2/paket/module2.py

C:\Users\dinoz\OneDrive\Рабочий стол\SE\laba16\laba16>git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\dinoz\OneDrive\Рабочий стол\SE\laba16\laba16>git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/InternetHacker1123/laba16.git
 c4a35de..5eb3809  main -> main

```

Рисунок 8.1 – Коммит файлов в репозитории git

Контрольные вопросы

1. Что является модулем языка Python?

Под модулем в Python понимается файл с расширением `.py`. Модули предназначены для того, чтобы в них хранить часто используемые функции, классы, константы и т. п. Можно условно разделить модули и программы:

программы предназначены для непосредственного запуска, а модули для импортирования их в другие программы. Стоит заметить, что модули могут быть написаны не только на языке Python, но и на других языках (например C).

2. Какие существуют способы подключения модулей в языке Python?

Самый простой способ импортировать модуль в Python это воспользоваться конструкцией: `import имя_модуля`. За один раз можно импортировать сразу несколько модулей, для этого их нужно перечислить через запятую после слова `import`. Если вы хотите задать псевдоним для модуля в вашей программе, можно воспользоваться вот таким синтаксисом: `import имя_модуля as новое_имя`. Используя любой из вышеперечисленных подходов, при вызове функции из импортированного модуля, вам всегда придется указывать имя модуля (или псевдоним). Для того, чтобы этого избежать делайте импорт через конструкцию `from ... import...`

3. Что является пакетом языка Python?

Пакет в Python – это каталог, включающий в себя другие каталоги и модули, но при этом дополнительно содержащий файл `__init__.py`. Пакеты используются для формирования пространства имен, что позволяет работать с модулями через указание уровня вложенности (через точку).

4. Каково назначение файла `__init__.py` ?

Файл `__init__.py` может быть пустым или может содержать переменную `__all__`, хранящую список модулей, который импортируется при загрузке через конструкцию.

5. Каково назначение переменной `__all__` файла `__init__.py` ?

Переменная `__all__` в файле `__init__.py` в Python хранит список модулей, который импортируется при загрузке через конструкцию