**Урок № 5, 6**

**Клас 10-Б**

**Дата уроку 10.09.2020**

**Тема. ПР1.1. Робота в інтерактивному режимі інтерпретатора IDLE. Ознайомлення з системою контролю версій Git**

**Мета. Навчальна**: Формування практичних навичок роботи в інтерактивному режимі з Python, ознайомлення з системою контролю версій

**Виховна**: виховання інтересу до вивчення предмету, потяг до наукової творчості, виховання позитивних рис характеру, сумлінності, здатності до переборення труднощів.

**Розвиваюча**: розвиток мислення, пам’яті, уваги.

**Тип уроку:** комбінований

**Структурні компоненти уроку**

|  |  |
| --- | --- |
| 1. Організація класу |  |
| 1. Актуалізація і корекція опорних знань |  |
| 1. Повідомлення теми, мети і завдань уроку |  |
| 1. Мотивація навчальної діяльності учнів |  |
| 1. Сприймання нового матеріалу, його усвідомлення. |  |
| 1. Застосування набутих знань про способи дії |  |
| 1. Узагальнення і систематизація знань. |  |
| 1. Підсумки уроку |  |
| 1. Домашнє завдання |  |

**Організація класу**

Привітання, відмічання відсутніх

**Актуалізація і корекція опорних понять та уявлень**

Питання для фронтального опитування учнів

1. Якими операційними системами підтримується мова Python?
2. Яке розширення мають файли програм, які створено в консольному режимі?
3. Назвіть основні переваги мови Python.
4. Який тип трансляції застосовується у Python?
5. Із якими мовами може інтегруватися мова Python?
6. Що називають динамічною типізацією даних?
7. Поясніть структуру проекту мовою Python.
8. Яку структуру може мати модуль мовою Python?
9. Для чого застосовується коментар у програмному коді?
10. Яке позначення має знак запрошення в інтерактивному режимі?
11. Як завершується робота інтерпретатора в інтерактивному режимі?
12. Які переваги має інтерактивний режим виконання програмного коду?
13. Як можна запустити інтерактивний режим інтерпретатора IDLE?
14. Поясніть сутність інтерактивного режиму.
15. Як правильно увести кілька команд в одному рядку?
16. Як оформлюється блок команд у мові Python?

**Повідомлення теми, мети і завдань уроку**

Повідомляється тема уроку

Разом з учнями визначається мета уроку

**Мотивація навчальної діяльності учнів**

Повідомляється значення даної теми в курсі науки. Звертається увага на її зв’язок з раніше вивченим та значення до матеріалу, який вивчатиметься в на наступних уроках.

**Сприймання нового матеріалу та його усвідомлення**

Системи контролю версій

<https://proglib.io/p/git-for-half-an-hour>

Ігрова система вивчення роботи з git <https://learngitbranching.js.org/?locale=uk>

**Застосування набутих знань про способи дії**

Інструктаж до виконання практичної роботи

Виконання практичної роботи

**Узагальнення та систематизація набутих знань**

1. Для чого призначено системи контролю версій?
2. Які онлайн сервіси для систем контрою версій ви знаєте?

**Підсумки уроку**

**Домашнє завдання**

Повторити за підручником §1.4

<https://proglib.io/p/git-for-half-an-hour>

<https://learngitbranching.js.org/?locale=uk>

Додаткові матеріали до уроку

<http://cs.mipt.ru/python/lessons/lab2.html#id3>

## [черепаха](http://cs.mipt.ru/python/lessons/lab2.html#id36)

Стандартна бібліотека Python містить модуль turtle, призначений для навчання програмуванню. Цей модуль містить [**набір**](https://docs.python.org/3/library/turtle.html#methods-of-rawturtle-turtle-and-corresponding-functions) функцій, що дозволяють управляти черепахою. Черепаха вміє виконувати невеликий набір команд, а саме:

| **команда** | **значення** |
| --- | --- |
| forward (X) | Пройти вперед X пікселів |
| backward (X) | Пройти назад X пікселів |
| left (X) | Повернутися наліво на X градусів |
| right (X) | Повернутися направо на X градусів |
| penup () | Не залишати слід при русі |
| pendown () | Залишати слід при русі |
| shape (X) | Змінити значок черепахи ( "arrow", "turtle", "circle", "square", "triangle", "classic") |
| stamp () | Намалювати копію черепахи в поточному місці |
| color () | встановити колір |
| begin\_fill () | Необхідно викликати перед малюванням фігури, яку треба зафарбувати |
| end\_fill () | Викликати після закінчення малювання фігури |
| width () | Тут можна встановити товщину лінії |
| goto (x, y) | Перемістити черепашку в точку (x, y) |

Наприклад, наступна програма малює букву S:

**import**  **turtle**

turtle . shape ( 'turtle' )

turtle . forward ( 50 )

turtle . left ( 90 )

turtle . forward ( 50 )

turtle . left ( 90 )

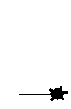
turtle . forward ( 50 )

turtle . right ( 90 )

turtle . forward ( 50 )

turtle . right ( 90 )

turtle . forward ( 50 )

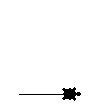


### [Вправа №2: буква S](http://cs.mipt.ru/python/lessons/lab2.html#id37)

Збережіть і виконайте попередню програму. Переконайтеся в тому, що черепаха працює.

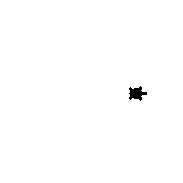
### [Вправа №3: ​​квадрат](http://cs.mipt.ru/python/lessons/lab2.html#id38)

Намалюйте квадрат. приклад:



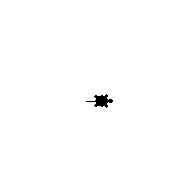
### [Вправа №4: коло](http://cs.mipt.ru/python/lessons/lab2.html#id39)

Намалюйте коло. Скористайтеся тим фактом, що правильний багатокутник з великим числом сторін буде виглядати як коло. приклад:



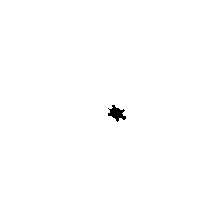
### [Вправа №5: більше квадратів](http://cs.mipt.ru/python/lessons/lab2.html#id40)

Намалюйте 10 вкладених квадратів.



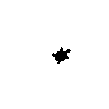
### [Вправа №6: павук](http://cs.mipt.ru/python/lessons/lab2.html#id41)

Намалюйте павука з n лапами. Приклад n = 12:



### [Вправа №7: спіраль](http://cs.mipt.ru/python/lessons/lab2.html#id42)

Намалюйте спіраль. Див. [**Теорію**](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D0%BC%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0_%D1%81%D0%BF%D0%B8%D1%80%D0%B0%D0%BB%D1%8C) . приклад:



### [Вправа №8: квадратна «спіраль»](http://cs.mipt.ru/python/lessons/lab2.html#id43)

Намалюйте «квадратну» спіраль. приклад:

