Lab 2 Task 3 Solution

3) Become familiar with function *main*

1. Where is a *NumberGame* constructed?

   - By observation, we can conclude a *NumberGame* constructed inside function *main*

```python
def main() -> None:
    ...
    while True:
        g = NumberGame(goal, minimum, maximum, (p1, p2)) #<-
    Here!!
        winner = g.play()
        print(f'And {winner} is the winner!!!')
        print(p1)
        print(p2)
        again = input('Again? (y/n) ')
        if again != 'y':
            return
```

2. This function calls *g.play* repeatedly in a loop. What about the game can change each time *g.play* is called: the goal, the min or max move, the players, the moves?

   - By observation, we can conclude that
     1. the goal doesn't change
     2. the min or max move don't change
     3. the current player change as a result of whose_turn method.

```python
def play(self) -> str:
    ...
    while self.current < self.goal:
        self.play_one_turn() # <- In here
    ...
    winner = self.whose_turn(self.turn - 1)
    return winner.namePlayers

def play_one_turn(self) -> None:
    ...
```

```
11              next_player = self.whose_turn(self.turn) # <-
        Here!!
12              amount = next_player.move(
13                  self.current,
14                  self.min_step,
15                  self.max_step,
16                  self.goal
17              )
18              self.current += amount
19              self.turn += 1
20
21              print(f'{next_player.name} moves {amount}.')
22              print(f'Total is now {self.current}.')
23
24
25          def whose_turn(self, turn: int) -> Player:
26              ...
27              if turn % 2 == 0:
28                  return self.players[0]
29              else:
30                  return self.players[1]
31
32
```

4. the move changes by the *move* method in *play_one_turn*.

```
1               def play(self) -> str:
2                   ...
3                   while self.current < self.goal:
4                       self.play_one_turn()
5                   ...
6                   winner = self.whose_turn(self.turn - 1)
7                   return winner.namePlayers
8
9               def play_one_turn(self) -> None:
10                  ...
11                  next_player = self.whose_turn(self.turn)
12                  amount = next_player.move( # <- Here!!
13                      self.current,
14                      self.min_step,
15                      self.max_step,
16                      self.goal
17                  )
18                  self.current += amount
19                  self.turn += 1
20
21                  print(f'{next_player.name} moves {amount}.')
22                  print(f'Total is now {self.current}.')
23
24
```

3. List all the places in this function where a *Player* is stored, an instance attribute of *Player* is accessed or set, or a method is called on a *Player*.

- We need to find all places in this function where *Player* is stored, where an instance attribute of *Player* is accessed or set, or where a method is called on a *Player*.

  First, we need to find where *Player* is stored in this function.

  The code tells us the type of third argument in *NumberGame* is Tuple, and each element in the tuple is taken by the variables *p1* and *p2*.

```
1           def main() -> None:
2               ...
3               while True:
4                   g = NumberGame(goal, minimum, maximum, (p1, p2)) #
    <- Here!!
5                   ...
6
```

  Because we know each element in the tuple is of type *Player*, we can conclude *Player* is stored inside variables *p1* and *p2*

  Second, we need to find where the instance instance attribute of *Player* is accessed or set in this function.

  By observation, no instance attribute of *Player* is accessed or set.

  Finally, we need to find where a method of *Player* is called in this function.

  By observation, no method of *Player* is accessed or set inside this function.