Chapter 17 숫자 맞추기 게임 만들기

날짜	@November 24, 2021
발표자	김윤서

17.1 해법

1. 0~99 사이의 랜덤한 숫자 하나 정하기
2. 사용자 입력 받기
3. 입력값과 랜덤값을 비교 후 입력 값이 큰지 작은지 출력하기. 다시 사용자 입력을 받아서 반복하기
4. 만약 숫자가 맞으면 맞췄다 + 시도횟수 출력하기.
5. 프로그램 종료하기

17.2 사전지식

랜덤 숫자 → rand.Intn(range) 함수 사용, 현재 시각을 인수로 주기

랜덤 함수를 지원하는 math/rand 패키지와 현재 시각을 지원하는 time 패키지 알아보기

17.2.1 math/rand 패키지

랜덤한 숫자를 얻으려면 math/rand 패키지를 임포트해야 한다. 특정 범위에서 int 타입 랜덤값을 생성하는 rand.Intn(range) 함수 사용. range 인수는 생성되는 값의 범위, 0보다 커야 함. 랜덤값은 0 ~ range-1 사이

```
func Intn(n int) int
```

완전 랜덤 X 유사 랜덤값 : 어떤 알고리즘에 의해 마치 랜덤처럼 보이는 값들, 컴퓨터의 논리회로와 산술 연산은 랜덤값을 만들기 적합하지 않다. 단순 rand.Intn() 함수는 매번 똑같은 값이 생

성되는 문제, 랜덤값이 산출되는 초깃값이 같기 때문이다. → 실행할 때마다 매번 랜덤 시드를 다른 값으로 설정

```
func Seed(seed int64)
```

17.2.2 time 패키지

time 패키지는 날짜, 시각, 시간 등을 다루는 패키지. 현재 시각 값을 랜덤 시드값으로 설정 → 매번 다른 랜덤값 생성

```
func Now() Time
```

Time 객체 → UnixNano() → int64값으로 변환

UnixNano(): UTC 시간 기준 1970년 1월 1일부터 Time 객체가 나타내는 시각까지 경과한 시간을 나노초 단위로 나타낸 값

```
func (t Time) UnixNano() int64
```

17.3 랜덤한 숫자 생성하기

```
package main

import (
  "fmt"
  "math/rand"
  "time"
)

func main() {
  rand.Seed(time.Now().UnixNano())

  n := rand.Intn(100)
  fmt.Println(n)
}
```

출력문

```
27
```

17.4 숫자값 입력받기

숫자값 입력받기 → fmt.Scan() 계열 함수 이용

숫자 대신 문자 입력 → 에러 반환 & 다시 입력 받기 → 표준 입력 스트림 비워주기

```
package main

import (
  "bufio"
  "fmt"
  "os"
)

var stdin = bufio.NewReader(os.Stdin)

func InputIntValue() (int, error) {
  var n int
  _, err := fmt.Scanln(&n)
  if err != nil {
    stdin.ReadString('\n')
  }
  return n, err
}

func main() {
  for {
    fmt.Printf("숫자값을 입력하세요:")
    n, err := InputIntValue()
    if err != nil {
      fmt.Println("숫자만 입력하세요.")
    } else {
      fmt.Println("입력하신 숫자는 ", n, " 입니다.")
    }
  }
}
```

출력문

```
숫자값을 입력하세요:ex
숫자만 입력하세요.
숫자값을 입력하세요:80
입력하신 숫자는  80  입니다.
숫자값을 입력하세요:3
입력하신 숫자는  3  입니다.
숫자값을 입력하세요:
```

- int 타입값을 읽는 데 숫자 아닌 문자 → 에러 반환 & 입력 스트림 비워주기 (stdin.ReadString('\n'))

17.5 숫자 맞추기 완성하기

```
package main

import (
  "bufio"
  "fmt"
  "math/rand"
  "os"
  "time"
)

var stdin = bufio.NewReader(os.Stdin)

func InputIntValue() (int, error) {
  var n int
  _, err := fmt.Scanln(&n)
  if err != nil {
    stdin.ReadString('\n')
  }
  return n, err
}

func main() {
  rand.Seed(time.Now().UnixNano())

  r := rand.Intn(100)
  cnt := 10
  for {
    fmt.Printf("숫자값을 입력하세요:")
    n, err := InputIntValue()
    if err != nil {
      fmt.Println("숫자만 입력하세요.")
    } else {
      if n > r {
        fmt.Println("입력하신 숫자가 더 큽니다.")
      } else if n < r {
        fmt.Println("입력하신 숫자가 더 작습니다.")
      } else {
        fmt.Println("숫자를 맞췄습니다. 축하합니다. 시도한 횟수:", cnt)
        break
      }
      cnt++
    }
  }
}
```

출력문

```
숫자값을 입력하세요:500
입력하신 숫자가 더 큽니다.
숫자값을 입력하세요:50
입력하신 숫자가 더 작습니다.
숫자값을 입력하세요:75
입력하신 숫자가 더 큽니다.
숫자값을 입력하세요:60
입력하신 숫자가 더 작습니다.
숫자값을 입력하세요:67
입력하신 숫자가 더 작습니다.
숫자값을 입력하세요:70
숫자를 맞췄습니다. 축하합니다. 시도한 횟수: 5
```

연습문제

1. Seed, UnixNano
2.

```
package main

import (
  "bufio"
  "fmt"
  "math/rand"
  "os"
  "time"
)

var stdin = bufio.NewReader(os.Stdin)

func InputIntValue() (int, error) {
  var n int
  _, err := fmt.Scanln(&n)
  if err != nil {
    stdin.ReadString('\n')
  }
  return n, err
}

func main() {
  rand.Seed(time.Now().UnixNano())

  balance := 1000

  for {
    r := rand.Intn(5) + 1
```

```
    fmt.Print("1~5사이의 값을 입력하세요:")
    n, err := InputIntValue()
    if err != nil {
      fmt.Println("숫자만 입력하세요.")
    } else if n < 1 || n > 5 {
      fmt.Println("1~5사이의 값만 입력하세요.")
    } else {
      if n == r {
        balance += 500
        fmt.Println("맞췄습니다! 남은 돈: ", balance)
        if balance >= 5000 {
          fmt.Println("게임 승리")
          break
        }
      } else {
        balance -= 100
        fmt.Println("틀렸습니다! 남은 돈: ", balance)
        if balance <= 0 {
          fmt.Println("게임 오버")
          break
        }
      }
    }
  }
}
```