Dokumentacja projektowa

**Rzut Monetą**

|  |  |  |
| --- | --- | --- |
| Dokument | Dokumentacja projektowa | |
| Temat | RZUT MONETĄ | |
| Wersja dokumentu | Data powstania wersji | Przyczyna powstania wersji |
| 1 | 07.01.2021 06:35:00 | Rozpoczęcie projektu |
| Autor dokumentu | Tomasz Cieślik | |
| Status | W realizacji | |
| Plik dokumentu | projekt - orzeł czy Reszka.docx | |

[1 Wymagania 3](#_Toc60894000)

[2 Etap 1 – Utworzenie wersji konsolowej programu 4](#_Toc60894001)

[2.1 Założenia 4](#_Toc60894002)

[3 RZUT MONETĄ 5](#_Toc60894003)

[3.1 Testy Jednostkowe 5](#_Toc60894004)

[4 Etap 3 – Utworzenie aplikacji z interfejsem graficznym 6](#_Toc60894005)

[5 Etap 4 – Dodatkowe prace 7](#_Toc60894006)

[6 8](#_Toc60894007)

[7 Instalacja – Wdrożenie 9](#_Toc60894008)

[7.1 Udostępnienie programu na Github.com 9](#_Toc60894009)

# Wymagania

1. Utworzenie programu mającego symulować rzut monetą.
2. Program ma być stworzony w wersji konsolowej oraz w wersji okienkowej.
3. Użytkownik decyduje ile monet jednocześnie bierze udział w losowaniu – **maksymalnie 5.**
4. Użytkownik może zmienić nazwę awersu i rewersu monety.

# Etap 1 – Utworzenie wersji konsolowej programu

## Założenia

1. Program ma być wykonany przy wykorzystaniu **C#** technologii **.Net Framework**
2. **Wersja nie uwzględnia zmiany**
3. Po uruchomieniu programu pojawia się pytanie „*Ile monet?* *– wpisz liczbę w przedziale od 1 do 5”.*
4. W przypadku braku wpisania liczby program automatycznie ma wylosować jedną monetę,
5. W przypadku wpisania liczby/ znaku spoza zdefiniowanego zakresu pojawia się komunikat:
   1. „Podano błędną wartość wpisz liczbę w przedziale od 1 do 5”.
6. Wyniki losowania mają zostać przedstawione w następujący sposób: (poniżej opisano strukturę poszczególnych wierszy):
   * 1. *Wylosowano: (pojawia się szczegółowa informacja o wyniku losowania np.: ORZEŁ – RESZKA – ORZEŁ --ORZEŁ*
     2. *ORZEŁ – Z ,*
     3. *RESZKA – Y .*
7. Program ma być zapętlony:
8. Po jego wykonaniu użytkownikowi wyświetla się następujące pytanie:
   1. Czy chcesz jeszcze raz zagrać? (wpisz ”**Nie**” jeśli chcesz zakończyć), następnie zatwierdź).

# RZUT MONETĄ

## Testy Jednostkowe

Automatyczne testy jednostkowe mają uwzględniać następujące przypadki testowe:

1. Losowanie:
2. 5 monet orzeł i reszka prawidłowo wprowadzonych
3. 1 moneta ze zmienionym opisem na Awers i Rewers – przypadek tworzony w trakcie realizacji etapu 4
4. Brak wprowadzonej ilości monet
5. Wprowadzona wartość cyfrowa spoza zdefiniowanego zakresu
6. Wprowadzony inny znak niż liczba
7. Kończenie działania programu:
   1. Wpisanie Tak – oczekiwane zamknięcie programu
   2. Wpisanie Nie – oczekiwanie dalsze działanie programu
   3. Wpisanie innej wartości – oczekiwanie ponownego pytania w sprawie zamknięcia programu.

# Etap 3 – Utworzenie aplikacji z interfejsem graficznym

W przygotowaniu.

# Etap 4 – Dodatkowe prace

W przygotowaniu.

1. Dodanie opcji zmiany Awersu i rewersu w obu wersjach
2. Prace nad wyglądem graficznym – ustawienie rozmiaru okna wielkości oraz położenia danych

# Instalacja – Wdrożenie

W przygotowaniu

## Udostępnienie programu na Github.com