

LU12.A03 - Lotto

Ausgangslage

Mit der Applikation „Lotto“ können Benutzer Ihr persönliches Lotto-Konto verwalten. Die Benutzer können:

- Ein- und Auszahlungen vornehmen
- Spieltipps mit ihren 6 Zahlen + Jokerzahl abgeben. Jeder Spieltipp kostet CHF 2.00

Screenshots

Passwort

```
Passwort > abcdefg
Passwort falsch
Passwort > passWORT
```

Hauptmenu

```
Lotto
-----

A) Konto Ein- und Auszahlungen tätigen
B) Lottotipps abgeben
Z) Beenden
Ihre Wahl > q

Bitte geben Sie eine gültige Wahl ein
Ihre Wahl >
```

Ein- / Auszahlungen

```
Aktueller Kontostand: 6.0
Auswahl (A, E oder Z) > E
Betrag Einzahlung > 5
Betrag muss zwischen 10.00 und 20.00 sein
Betrag Einzahlung > 3a
Geben Sie eine Zahl ein
Betrag Einzahlung > 12
Neuer Kontostand: 18.0
Auswahl (A, E oder Z) >
```

Lottoschein

1. Zahl: Geben Sie eine Zahl von 1 bis 42 ein > a
Bitte geben Sie eine Ganzzahl ein
1. Zahl: Geben Sie eine Zahl von 1 bis 42 ein > 12
2. Zahl: Geben Sie eine Zahl von 1 bis 42 ein > 2.4
Bitte geben Sie eine Ganzzahl ein
2. Zahl: Geben Sie eine Zahl von 1 bis 42 ein > 1
3. Zahl: Geben Sie eine Zahl von 1 bis 42 ein > 12
Bitte geben Sie eine Zahl ein, die Sie nicht schon gewählt haben
3. Zahl: Geben Sie eine Zahl von 1 bis 42 ein > 4
4. Zahl: Geben Sie eine Zahl von 1 bis 42 ein > 9
5. Zahl: Geben Sie eine Zahl von 1 bis 42 ein > 40
6. Zahl: Geben Sie eine Zahl von 1 bis 42 ein > 23

Jokerzahl: Geben Sie eine Zahl von 1 bis 6 ein > 5

X	2	3	X	5	6
7	8	X	10	11	X
13	14	15	16	17	18
19	20	21	22	X	24
25	26	27	28	29	30
31	32	33	34	35	36
37	38	39	X	41	42


Jokerzahl: 5

Neuer Kontostand: 18.0

Vorgehen

Bei diesem Programm arbeiten Sie als Team zusammen via GitHub. Git ist ziemlich gut darin, die Codeteile der verschiedenen Mitarbeiter zusammen zu bringen. Trotzdem braucht es ein wenig Koordination zwischen den Mitarbeitern.

Vorbereitung

1. Akzeptieren Sie das Assignment im GitHub Classroom.
2. Sie können die Gruppen (6 Gruppen à 3-4 Personen) selber bilden:
 - Entweder: Sie erstellen eine neue Gruppe.
 - Oder: Sie wählen eine bestehende Gruppe aus.
3. Setzen Sie sich mit Ihrer Gruppe an einen 4er-Tisch. Es ist **keine** Kommunikation über die Aufgabe ausserhalb Ihrer Gruppe erlaubt.
4. Klonen Sie das Repository in PyCharm.
5. Um die Issues zu erhalten, trage deinen Namen ins main.py ein und commit und push 
6. Überprüfe dein Repo ob die Issues vorhanden sind (dauert etwa 2 Minuten, Seite aktualisieren nicht vergessen)

- Erstellen Sie eine neue Python-Datei `my_test`. Falls PyCharm nachfragt, ob es die Datei mit Git verwalten soll, **Lehnen Sie dies ab**.
- Kopieren Sie den unten stehenden Code in diese Datei.

my_test

```
# TODO import(s)

if __name__ == '__main__':
    # TODO create objects if needed
    # TODO call the function
    # TODO print the result (if applicable)
    pass
```

Die Datei `my_test` hilft Ihnen, ihre Funktionen und Klassen zu testen.



- Fügen Sie einen `import`-Befehl für das Modul bzw. die Klasse ein, die Sie testen möchten.
- Fügen Sie den Aufruf Ihrer Funktion ein
 - Übergeben Sie allfällige Parameter.
 - Geben Sie einen allfälligen Returnwert mittels `print(...)` in der Konsole aus.

Realisierung



- Suchen Sie in der Übersicht diejenigen Tasks,
 - die noch von keinem Teammitglied bearbeitet werden,
 - deren Vorgänger bereits abgeschlossen sind.
- Wählen Sie einen dieser offenen Task (Issue) aus.
- Tragen Sie sich als zuständige Person ein: Issue auswählen und `assign yourself` klicken.
- Aktualisieren Sie den Stand Ihres Projekts in PyChar mittels **Update Project**
 - Lösen Sie allfällige Merge Konflikte
- Codieren Sie die Klasse oder Funktion schrittweise. Immer wenn Sie einen Stand erreicht haben, bei dem ihr Code
 - lauffähig ist und
 - einen gültigen Returnwert liefert
 - Commit der Änderung inkl. sinnvoller Commit Message
 - Übernehmen Sie die Änderungen Ihrer Kollegen mittels **Update Project**
 - Lösen Sie allfällige Merge Konflikte
 - Push auf GitHub
- Falls die Klasse / Funktion noch nicht vollständig ist, gehen Sie wieder zu Schritt 3
- Ändern Sie den Label des Issues von `open` zu `ready for test`

Der Ablauf um eine Änderung ins GitHub Repository zu transferieren ist immer:



1. **Commit** (Festschreiben der Änderung im lokalen Repository auf Ihrem Computer).
2. **Update Project** bzw. **Pull** (Lokales Repository mit den neuesten Änderungen von GitHub aktualisieren).
3. **Push** (Aktualisieren des Repositories auf GitHub).

Testen

Unit Tests

Beim Unit Test wird eine einzelne Funktion getestet. Dabei sollen Sie gegenseitig versuchen, Fehler in den Funktionen der anderen Teammitglieder zu finden.

1. Wählen Sie eine fertig programmierte Funktion eines Kollegen aus. Diese Issues haben den Label `ready for test`.
2. Tragen Sie sich im Issue auf GitHub als zusätzlicher Mitarbeiter ein. Ändern Sie den Label zu `testing`.
3. Aktualisieren Sie das Projekt in PyCharm mittels **Update Project**
4. Passen Sie Ihre Datei `my_test` so an, dass Sie die entsprechende Funktion importieren und aufrufen.
5. Testen Sie die Funktion mit ungewöhnlichen und kreativen Eingaben bzw. Werten.

Falls Sie einen Fehler entdecken, ergänzen Sie das Issue auf GitHub mit einem Kommentar. Diese Kommentar muss enthalten:

- Parameter die beim Aufruf der Funktion mitgegeben wurden
- Eingaben des Benutzers
- Erwartetes Resultat
- Effektives Resultat
- Allfällige Bemerkungen

Ändern Sie den Label des Issues zu `bug`.

Integrationstest

Führen Sie als Team mehrere Testläufe des Programms durch. Legen Sie vor den Testläufen fest, welches Resultat erscheinen soll. Halten Sie allfällige Abweichungen fest.

⇒ *GitHub Repo für externe Besucher*

GitHub Repository <https://github.com/templates-python/m319-lu12-a03-lottery>

Lernende am BZZ müssen den Link zum GitHub Classroom Assignment verwenden

M319-LU12



Marcel Suter

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/de/modul/m319/learningunits/lu12/aufgaben/lotto?rev=1750657505>

Last update: **2025/06/23 07:45**

