# Guess the day ver3.

|  |  |
| --- | --- |
| ver0 | original description |
| ver1 | finding class, attribute and method candidates |
| ver2 | Refining the public interface, refining attributes, methods, classes |
| ver3 | Defining more methods and attrivutes |

## Original description

Select-the-day, the program shows that day’s weekday and three other randomly picked week days, asks a user to select which day is today, checks the answer and saves the information to data storage accessible to the personnel of the service home. If the user picked the wrong day, the program asks again giving a gentle hint e.g. today is the first day of the week. And if the user guesses wrong the second time, the program shows a nice greeting and sets the game as FAILED to the data storage. If the user answered correctly the second time the game is saved as FOLLOW, and if the answer was right the first time, the game is saved as PASSED. The personnel checks the saved games to verify if the resident has problems in keeping up with the time.

## 1st iteration, check nouns.

**Select-the-day, the program shows that day’s weekday and three others randomly picked week days, asks a user to select which day is today, checks the answer and saves the information to data storage accessible to the personnel of the service home. If the user picked the wrong day, the program asks again giving a gentle hint e.g. today is the first day of the week. And if the user** **guesses wrong the second time, the program shows a nice greeting and sets the game as FAILED to the data storage. If the user answered correctly the second time the game is saved as FOLLOW, and if the answer was right the first time, the game is saved as PASSED. The personnel checks the saved games to verify if the resident has problems in keeping up with the time.**

|  |  |  |  |
| --- | --- | --- | --- |
| noun | basic form, synonym, duplicate value, notes | value or collections | possible class, object, attribute |
| **Select-the-day** |  | value | **Possible class:**  game’s title, game class, knows the rules of the game and keeps track of the game’s data |
| **weekdays** | weekday | value | **Possible attribute:**  Stores weekdays |
| **user** |  | type instance | **Possible attribute:**  Stores user instance |
| **today** |  | value | **Possible attribute:**  Stores current weekday |
| **answer** |  | Type string | **Possible attribute:**  Keeps value of user’s answers |
| **gentle hint** |  | type list, value | **Possible attribute:**  Stores hints user are given in different states of the game |
| **state** |  | Type string | **Possible attribute:** Keeps count of user attempts |
| **options** |  | type list | **Possible attribute:**  Stores days user can select from |
| **Tries** | Try | type integer | **Possible attribute:**  Keeps count of user tries to answer correctly |

## 2nd iteration, check verbs

**Select-the-day, the program shows that day’s weekday and three other randomly picked week days, asks a user to select which day is today, checks the answer and saves the information to data storage accessible to the personnel of the service home. If the user picked the wrong day, the program asks again giving a gentle hint e.g. today is the first day of the week. And if the user** **guesses wrong the second time, the program shows a nice greeting and sets the game as FAILED to the data storage. If the user answered correctly the second time the game is saved as FOLLOW, and if the answer was right the first time, the game is saved as PASSED. The personnel checks the saved games to verify if the resident has problems in keeping up with the time.**

|  |  |  |
| --- | --- | --- |
| verb | subject – object (who does, who is the target) | possible action, function, method |
| **checks** | Game’s action | **Possible method:**  Check is user answer correct, return True if it is, Else False |
| **saves** | Games’s action | **Possible method:**  Saves user’s answer to be accessed by service home personnel |
| **set state** | Games’s action | **Possible method:**  Sets game state according to user answer which is later saved in data storage |
| **random generate** | Games’s action | **Possible method:**  Randomly select three other weekdays + add correct weekday into mix |
| **quit** | Games’s action | **Possible method:**  Quits the game |
| **restart** | Games’s action | **Possible method:**  Restart the game |
| **set hint** | Games’s action | **Possible method:**  Check user answer and gives hint based on today’s weekday |

## 3rd iteration

@ver3 description

Select the day game asks user to select from 4 different options which weekday is today. If user answers incorrectly the first time, user is given gentle hint which weekday is today. If user answers wrong the second time, user is given nice greeting and state is saved as “FAILED”. If user answers correctly on one of the times, he gets message “PASSED” and state is saved as “PASSED”. Authorized staff will be able to access answer state of particular user in later time.

### Description

@Ver 1, @ver 2, @ver 3

1. class SelectTheDay
   1. Attributes
      1. Title
         1. Stores title of the game
      2. Weekdays
         1. Stores all weekdays.
      3. Today
         1. Stores today’s weekday
      4. answer
         1. Keeps value of user’s answers
      5. Options
         1. Randomly select three other weekdays + add correct weekday into mix
      6. State
         1. Stores three states which are used to keep track of current state of the game.
      7. Hints
         1. Stores hints user are given in different states of the game.
      8. Tries
         1. Keeps count of user tries to answer correctly.
   2. Methods
      1. Check
         1. Check is user answer correct, return True if it is, Else False
      2. Save
         1. Saves user’s answer to be accessed by staff later in time
      3. Set state.
         1. Sets game state according to user answer which is later saved in data storage
      4. Random generate.
         1. Randomly select three other weekdays + add correct weekday into mix
      5. Restart
         1. Restarts the game.
      6. Quit
         1. quits the game.
      7. Set hint.
         1. Check user answer and gives hint based on today’s weekday.
   3. Reserve topic
   4. Reserve topic

### Generated UML class diagram (<https://app.genmymodel.com/>) @ver 1, @ver 2, @ver 3.

Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

### Code generated from the UML diagram @ver1, @ver2, @ver 3.

class SelectTheDay(object):

    def \_\_init\_\_(self):

        self.title = ""

        self.hints = None

        self.weekdays = False

        self.today =

        self.asnwer = False

        self.options = False

        self.state =

        self.tries = False

    # Start of user code -> properties/constructors for SelectTheDay class

    # End of user code

    def check(self):

        # Start of user code protected zone for check function body

        raise NotImplementedError

        # End of user code

    def save(self):

        # Start of user code protected zone for save function body

        raise NotImplementedError

        # End of user code

    def set\_state(self):

        # Start of user code protected zone for set\_state function body

        raise NotImplementedError

        # End of user code

    def random\_generate(self):

        # Start of user code protected zone for random\_generate function body

        raise NotImplementedError

        # End of user code

    def restart(self):

        # Start of user code protected zone for restart function body

        raise NotImplementedError

        # End of user code

    def quit(self):

        # Start of user code protected zone for quit function body

        raise NotImplementedError

        # End of user code

    # Start of user code -> methods for SelectTheDay class

Completed code @ver 1, @ver 2, @ver 3

# File:         selecttheday.py

# Author(s): XXXXXX XXXXXX, XXXXXX XXXXXX, XXXXXX XXXXXX, XXXXXX XXXXXX

# CO-Author(s):    Pavel Kaljunen, Sebastian Sopola, Uras Ayanoglu, Jerry Karkainen

# Description:  This game is memory game. It is played to check if user gets day's weekday correct.

# -------------------------------------------------------------------------------------------------------------------------------

# Import necessary libaries

from datetime import datetime

import random

# -------------------------------------------------------------------------------------------------------------------------------

# This class handles game interaction

class SelectTheDay:

    def \_\_init\_\_(self):

        self.title = "Select-the-day"

        self.hints = ["Unfortunately this time you didn't guess it right.", "Today is first day of the week", "Today is second day of the week", "Today is third day of the week"]

        self.weekdays = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]

        self.today = datetime.datetime.now().strftime("%A")

        self.answer = ""

        self.options = ["","","",""]

        self.state = ""

        self.tries = 0

# Hints voisi teha sillai automaatioksi, etta yksi arvo on, ["Today is {datetime.datetime.now()} th day of the week"] muotoa.

# -----------------------------------------------------------------------------------------------------------------------------

    def check(self, answer): #ver2

        # Check user answer

        if answer == self.today:

            return True

        else:

            self.tries += 1

            return False

    def generate\_options(self):

        # randomly generate three wrong weekdays + add correct one to list options

        self.set\_hint(self.hint)

        self.weekdays.remove(self.today) #remove today from list to avoid duplicates in options

        self.options = random.sample(self.weekdays,3)

        self.options.append(self.today)

        random.shuffle(self.options)

    def set\_state(self, tries):

        # Sets game state according to user answer which is later saved in data storage

        if self.tries == 1:

            self.state = "FOLLOW"

        elif self.tries == 2:

            self.state = "FAILED"

            self.greet()

    def set\_hint(self, hint):

        # Possible method:Check user answer and gives hint based on today s weekday

        if self.today == "Monday":

            self.hint = "Today is the first day of the week"

        if self.today == "Tuesday":

            self.hint = "Today is the second day of the week"

        if self.today == "Wednesday":

            self.hint =  "Today is the third day of the week"

        if self.today == "Thursday":

            self.hint =  "Today is the fourth day of the week"

        if self.today == "Friday":

            self.hint =   "Today is the fifth day of the week"

        if self.today == "Saturday":

            self.hint =  "Today is the sixth day of the week"

        if self.today == "Sunday":

            self.hint =  "Today is the seventh day of the week"

    def save(self):

        # Saves user s answer to be accessed by staff

        pass

    def restart(self):

        # Start of user code protected zone for restart function body

        self.today = datetime.datetime.now().strftime("%A")

        self.options = []

        self.state = "PASSED"

        self.hint = ""

        self.tries = 0

    def quit(self):

        # User can quit game

        pass

# -------------------------------------------------------------------------------------------------------------------------