BSc (Hons) Artificial Intelligence and Data Science

Module: CM1601 Programming

Fundamentals

Individual Coursework Report

Module Leader: Sachinthani Perera

RGU Student ID	:	2330955
IIT Student ID	:	20231934
Student Name	:	Samed Ahamed Shouqi Mohamed Saqib

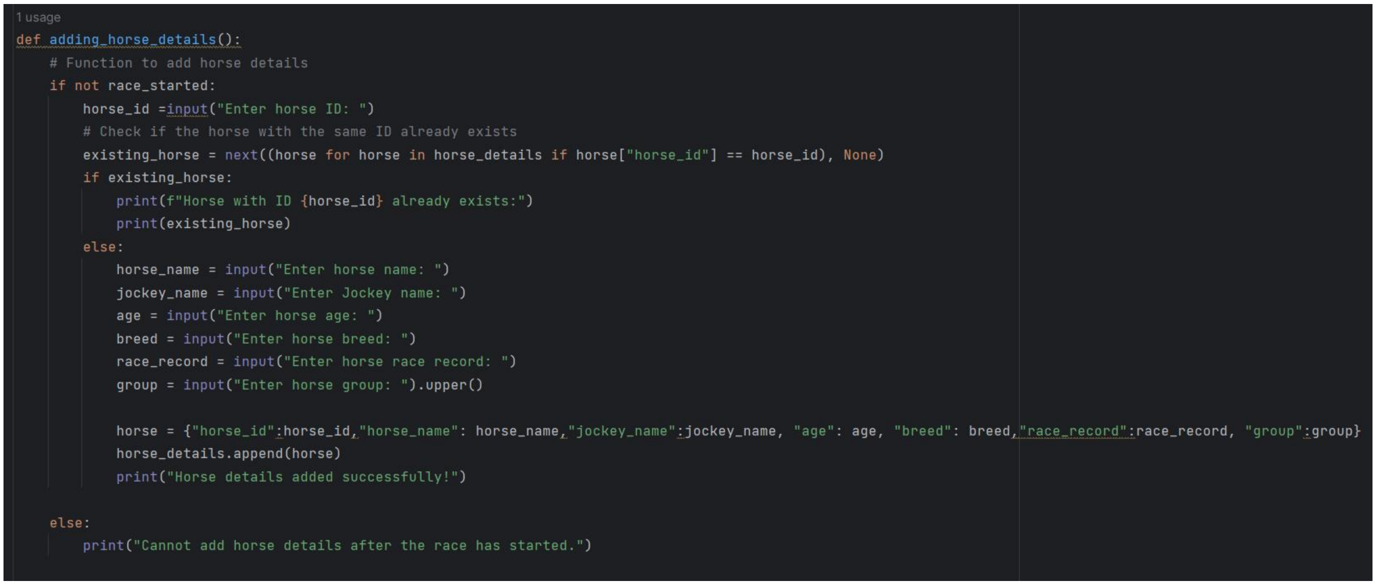
Table of Contents

1. Function: adding_horse_details()

Description:

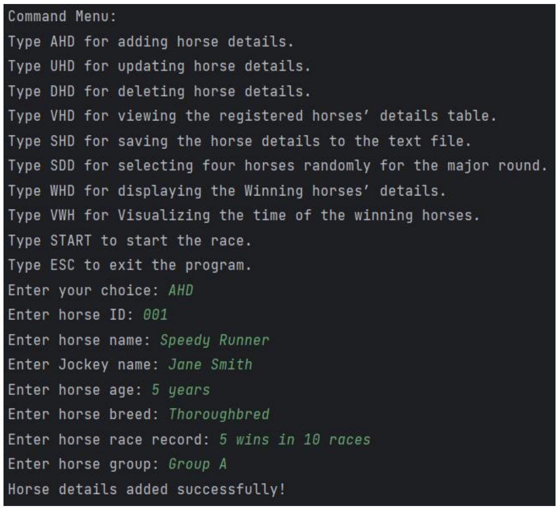
This function facilitates the addition of horse details to the system. It prompts the user to input essential information such as horse name, jockey name, age, breed, race record, and group. The function ensures that no two horses share the same ID, providing a warning if the ID is already in use. Once race is started, adding horse details, are restricted to maintain the integrity of the race.

Code:



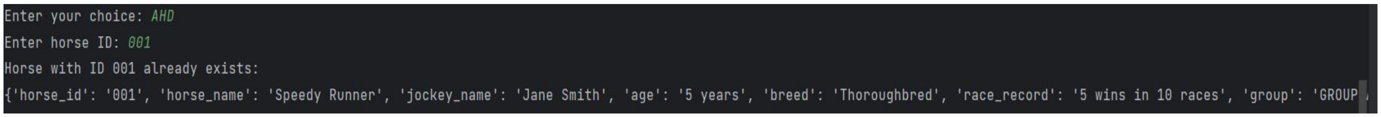
```python
1 usage
def adding_horse_details():
    # Function to add horse details
    if not race_started:
        horse_id =input("Enter horse ID: ")
        # Check if the horse with the same ID already exists
        existing_horse = next((horse for horse in horse_details if horse["horse_id"] == horse_id), None)
        if existing_horse:
            print(f"Horse with ID {horse_id} already exists:")
            print(existing_horse)
        else:
            horse_name = input("Enter horse name: ")
            jockey_name = input("Enter Jockey name: ")
            age = input("Enter horse age: ")
            breed = input("Enter horse breed: ")
            race_record = input("Enter horse race record: ")
            group = input("Enter horse group: ").upper()

            horse = {"horse_id":horse_id,"horse_name": horse_name,"jockey_name":jockey_name, "age": age, "breed": breed,"race_record":race_record, "group":group}
            horse_details.append(horse)
            print("Horse details added successfully!")

    else:
        print("Cannot add horse details after the race has started.")
```

Result:



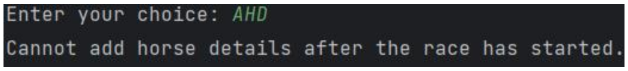
```
Command Menu:
Type AHD for adding horse details.
Type UHD for updating horse details.
Type DHD for deleting horse details.
Type VHD for viewing the registered horses’ details table.
Type SHD for saving the horse details to the text file.
Type SDD for selecting four horses randomly for the major round.
Type WHD for displaying the Winning horses’ details.
Type VWH for Visualizing the time of the winning horses.
Type START to start the race.
Type ESC to exit the program.
Enter your choice: AHD
Enter horse ID: 001
Enter horse name: Speedy Runner
Enter Jockey name: Jane Smith
Enter horse age: 5 years
Enter horse breed: Thoroughbred
Enter horse race record: 5 wins in 10 races
Enter horse group: Group A
Horse details added successfully!
```

Case 1: Adding Horse Details using a new horse id before the race starts



```
Enter your choice: AHD
Enter horse ID: 001
Horse with ID 001 already exists:
{'horse_id': '001', 'horse_name': 'Speedy Runner', 'jockey_name': 'Jane Smith', 'age': '5 years', 'breed': 'Thoroughbred', 'race_record': '5 wins in 10 races', 'group': 'GROUP
```

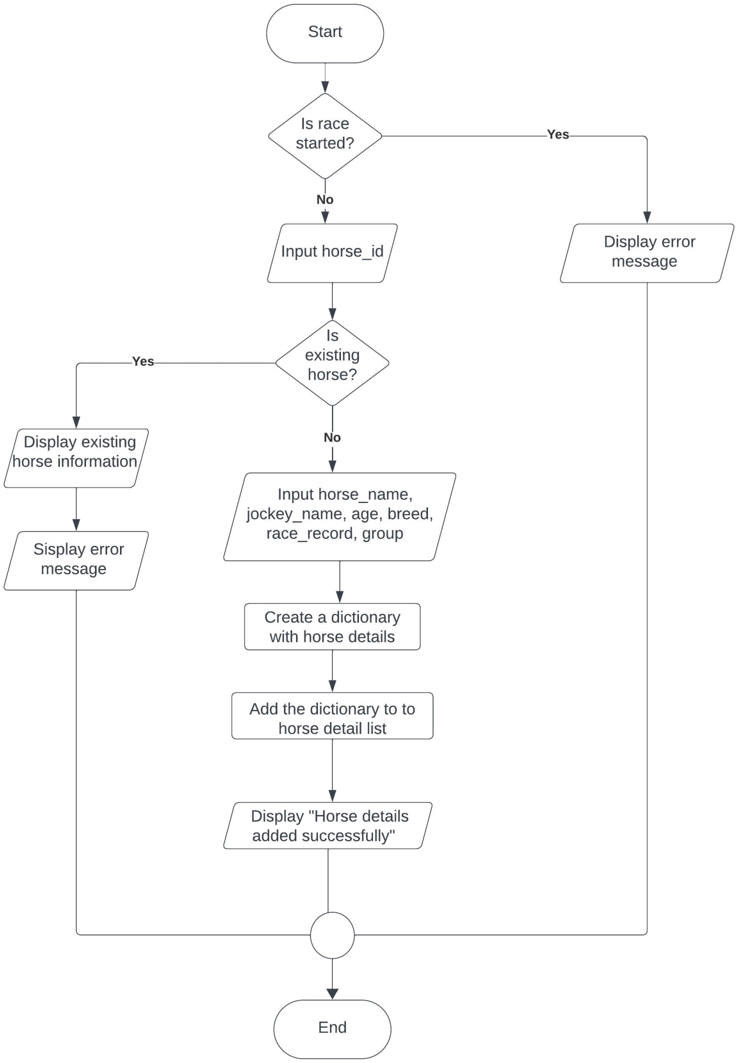
Case 2: Error message when trying to add horse details using an existing horse id



```
Enter your choice: AHD
Cannot add horse details after the race has started.
```

Case 3: Error message when trying to add horse details after the race has started

Flowchart:

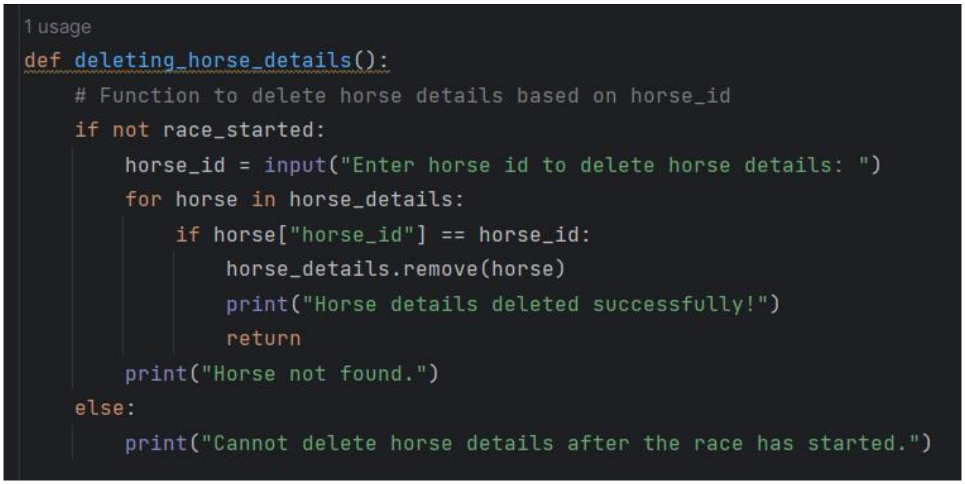

Start
Is race started?
Yes
No
Input horse_id
Display error message
Is existing horse?
Yes
No
Display existing horse information
Sisplay error message
Input horse_name, jockey_name, age, breed, race_record, group
Create a dictionary with horse details
Add the dictionary to to horse detail list
Display "Horse details added successfully"
End

2. Function: deleting_horse_details()

Description:

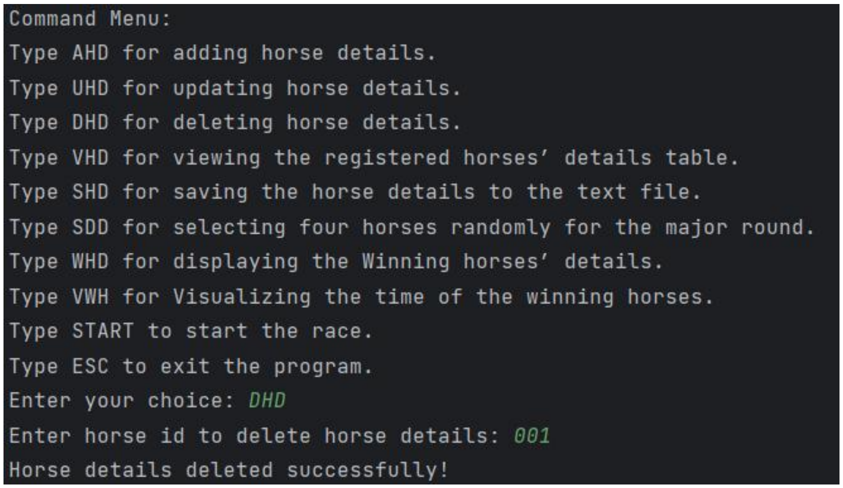
This function allows the removal of horse details based on the provided horse ID. Users are prompted to enter the ID of the horse they wish to delete. If the horse is found in the system, its details are removed; otherwise, a notification is displayed. Once race is started, deleting horse details, are restricted to maintain the integrity of the race.

Code:



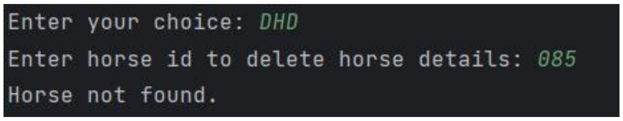
```
1 usage
def deleting_horse_details():
    # Function to delete horse details based on horse_id
    if not race_started:
        horse_id = input("Enter horse id to delete horse details: ")
        for horse in horse_details:
            if horse["horse_id"] == horse_id:
                horse_details.remove(horse)
                print("Horse details deleted successfully!")
                return
        print("Horse not found.")
    else:
        print("Cannot delete horse details after the race has started.")
```

Result:



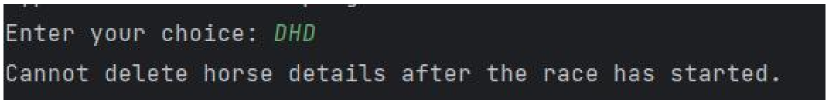
```
Command Menu:
Type AHD for adding horse details.
Type UHD for updating horse details.
Type DHD for deleting horse details.
Type VHD for viewing the registered horses’ details table.
Type SHD for saving the horse details to the text file.
Type SDD for selecting four horses randomly for the major round.
Type WHD for displaying the Winning horses’ details.
Type VWH for Visualizing the time of the winning horses.
Type START to start the race.
Type ESC to exit the program.
Enter your choice: DHD
Enter horse id to delete horse details: 001
Horse details deleted successfully!
```

Case 1: Deleting Horse Details using a horse id in horse_detail list before the race starts



```
Enter your choice: DHD
Enter horse id to delete horse details: 085
Horse not found.
```

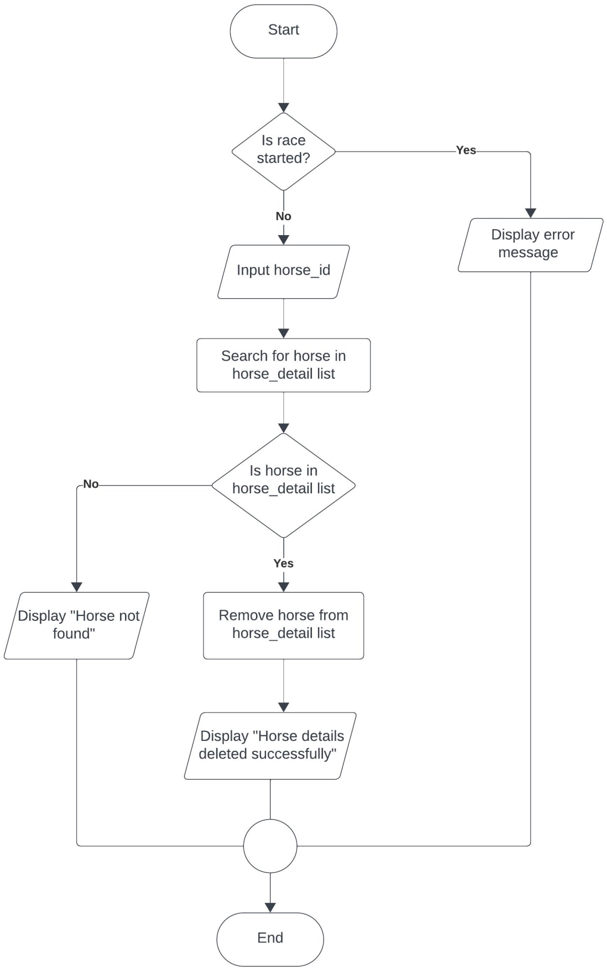
Case 2: Error message when trying to delete horse details using a horse id not in the horse_detail list



```
Enter your choice: DHD
Cannot delete horse details after the race has started.
```

Case 3: Error message when trying to delete horse details after the race has started

Flowchart:

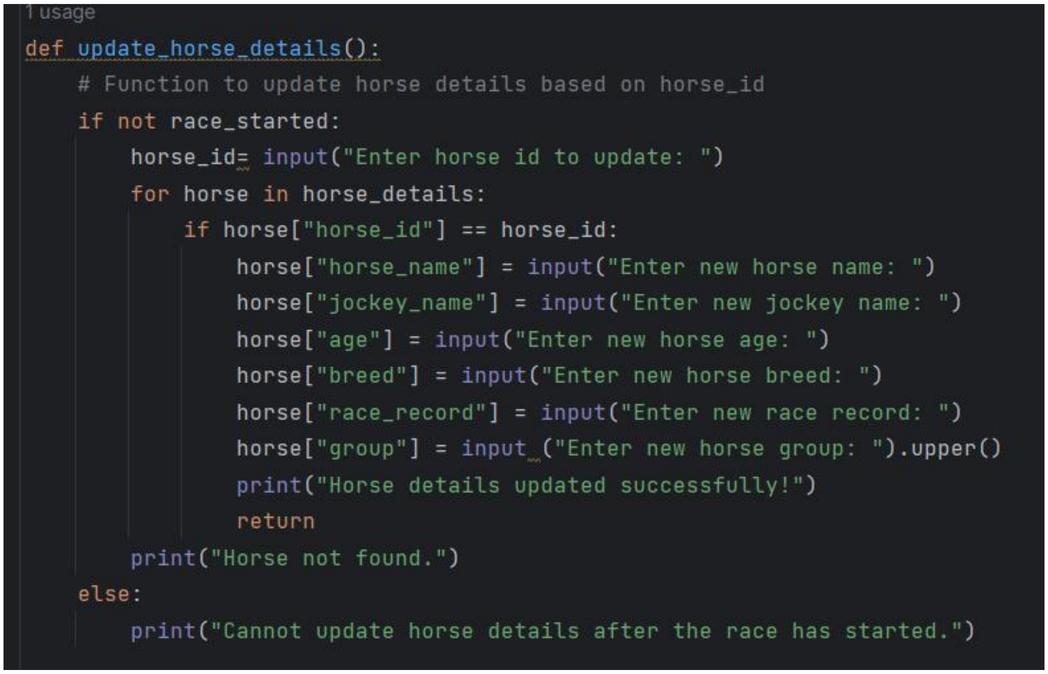

Start
Is race started?
Yes
No
Display error message
Input horse_id
Search for horse in horse_detail list
Is horse in horse_detail list
No
Yes
Display "Horse not found"
Remove horse from horse_detail list
Display "Horse details deleted successfully"
End

3. Function: update_horse_details()

Description:

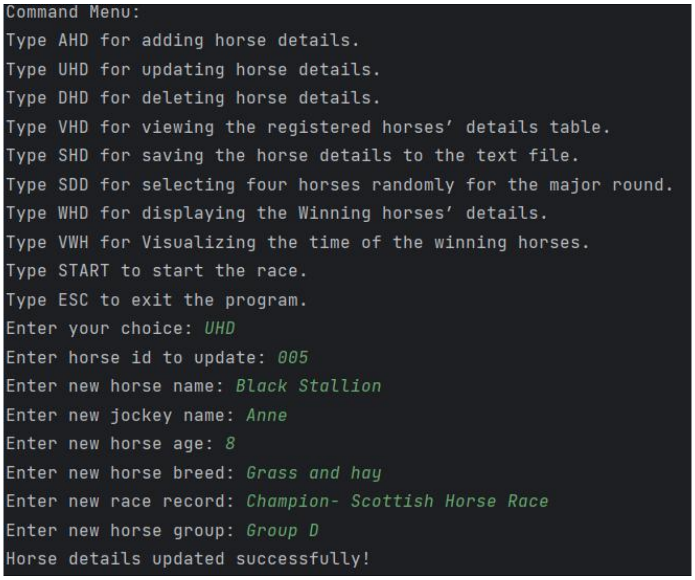
This function facilitates the modification of horse details based on the horse ID. Users are prompted to enter the ID of the horse they wish to update, and they can provide new information for the horse's name, jockey name, age, breed, race record, and group. Once race is started, updating horse details, are restricted to maintain the integrity of the race.

Code:



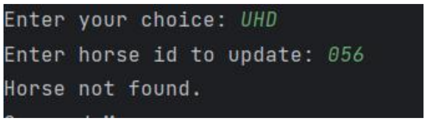
```python
def update_horse_details():
    # Function to update horse details based on horse_id
    if not race_started:
        horse_id= input("Enter horse id to update: ")
        for horse in horse_details:
            if horse["horse_id"] == horse_id:
                horse["horse_name"] = input("Enter new horse name: ")
                horse["jockey_name"] = input("Enter new jockey name: ")
                horse["age"] = input("Enter new horse age: ")
                horse["breed"] = input("Enter new horse breed: ")
                horse["race_record"] = input("Enter new race record: ")
                horse["group"] = input ("Enter new horse group: ").upper()
                print("Horse details updated successfully!")
                return
        print("Horse not found.")
    else:
        print("Cannot update horse details after the race has started.")
```

Result:



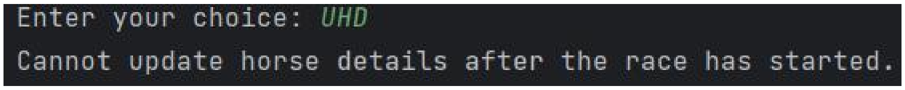
```
Command Menu:
Type AHD for adding horse details.
Type UHD for updating horse details.
Type DHD for deleting horse details.
Type VHD for viewing the registered horses’ details table.
Type SHD for saving the horse details to the text file.
Type SDD for selecting four horses randomly for the major round.
Type WHD for displaying the Winning horses’ details.
Type VWH for Visualizing the time of the winning horses.
Type START to start the race.
Type ESC to exit the program.
Enter your choice: UHD
Enter horse id to update: 005
Enter new horse name: Black Stallion
Enter new jockey name: Anne
Enter new horse age: 8
Enter new horse breed: Grass and hay
Enter new race record: Champion- Scottish Horse Race
Enter new horse group: Group D
Horse details updated successfully!
```

Case 1 : Updating Horse Details using a horse id in horse_detail list before the race starts



```
Enter your choice: UHD
Enter horse id to update: 056
Horse not found.
```

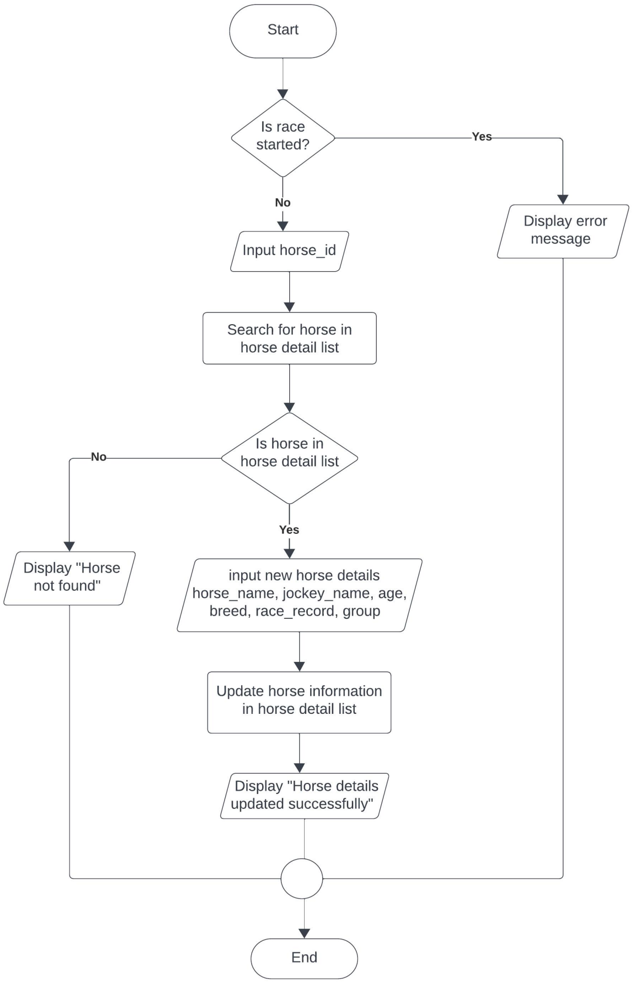
Case 2: Error message when trying to update horse details using a horse id not in the horse_detail list



```
Enter your choice: UHD
Cannot update horse details after the race has started.
```

Case 3: Error message when trying to update horse details after the race has started

Flowchart:

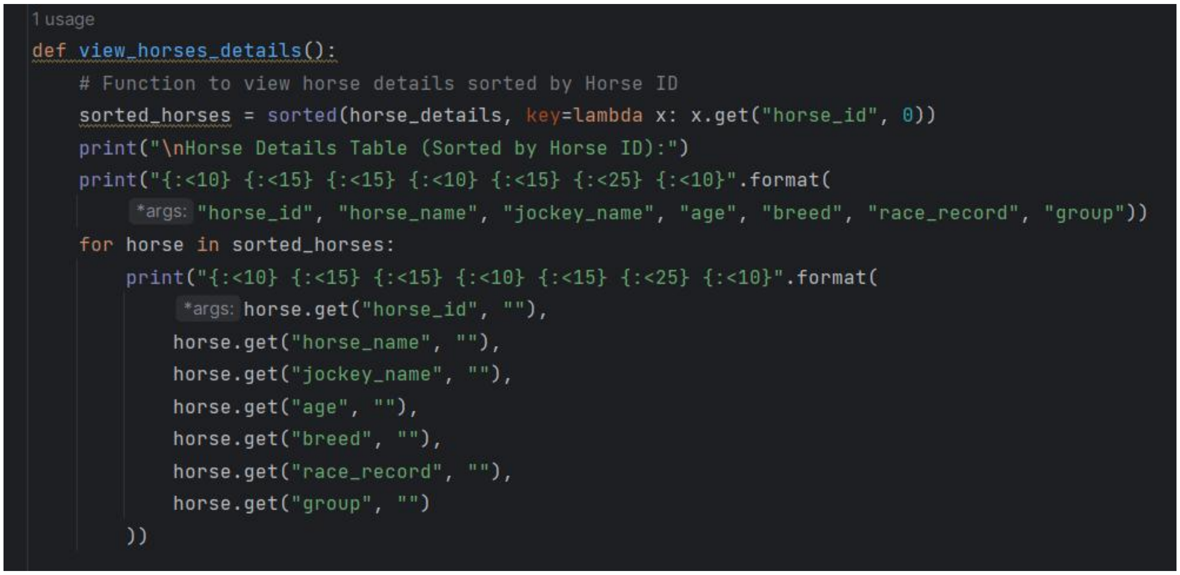

Start
Is race started?
Yes
Display error message
No
Input horse_id
Search for horse in horse detail list
Is horse in horse detail list
No
Display "Horse not found"
Yes
input new horse details horse_name, jockey_name, age, breed, race_record, group
Update horse information in horse detail list
Display "Horse details updated successfully"
End

4. Function: view_horses_details()

Description:

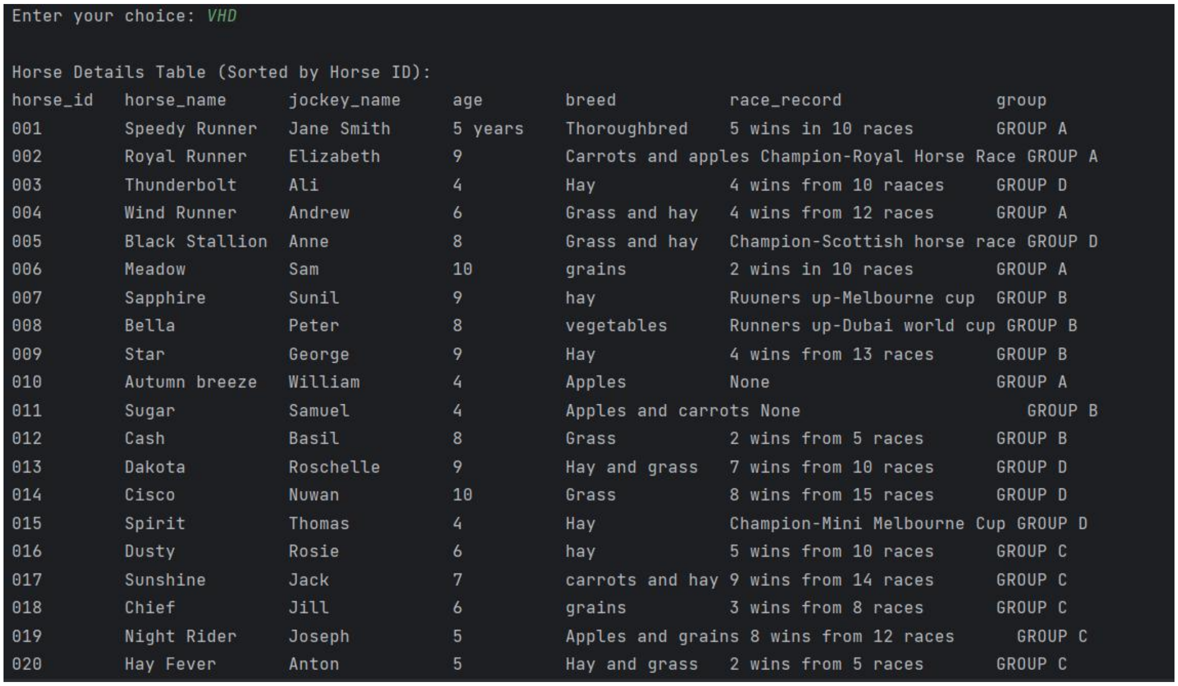
This function displays a table of horse details, sorted by Horse ID. The information presented includes horse ID, horse name, jockey name, age, breed, race record, and group. The table is formatted for easy readability.

Code:



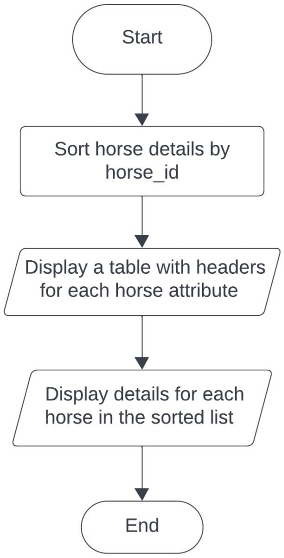
```
1 usage
def view_horses_details():
    # Function to view horse details sorted by Horse ID
    sorted_horses = sorted(horse_details, key=lambda x: x.get("horse_id", 0))
    print("\nHorse Details Table (Sorted by Horse ID):")
    print("{:<10} {:<15} {:<15} {:<10} {:<15} {:<25} {:<10}".format(
        *args: "horse_id", "horse_name", "jockey_name", "age", "breed", "race_record", "group"))
    for horse in sorted_horses:
        print("{:<10} {:<15} {:<15} {:<10} {:<15} {:<25} {:<10}".format(
            *args: horse.get("horse_id", ""),
            horse.get("horse_name", ""),
            horse.get("jockey_name", ""),
            horse.get("age", ""),
            horse.get("breed", ""),
            horse.get("race_record", ""),
            horse.get("group", "")
        ))
```

Result:



```
Enter your choice: VHD

Horse Details Table (Sorted by Horse ID):
horse_id   horse_name      jockey_name     age        breed           race_record               group
001        Speedy Runner   Jane Smith      5 years    Thoroughbred    5 wins in 10 races        GROUP A
002        Royal Runner    Elizabeth       9          Carrots and apples Champion-Royal Horse Race GROUP A
003        Thunderbolt     Ali             4          Hay             4 wins from 10 raaces     GROUP D
004        Wind Runner     Andrew          6          Grass and hay   4 wins from 12 races      GROUP A
005        Black Stallion  Anne            8          Grass and hay   Champion-Scottish horse race GROUP D
006        Meadow          Sam             10         grains          2 wins in 10 races        GROUP A
007        Sapphire        Sunil           9          hay             Ruuners up-Melbourne cup  GROUP B
008        Bella           Peter           8          vegetables      Runners up-Dubai world cup GROUP B
009        Star            George          9          Hay             4 wins from 13 races      GROUP B
010        Autumn breeze   William         4          Apples          None                      GROUP A
011        Sugar           Samuel          4          Apples and carrots None                      GROUP B
012        Cash            Basil           8          Grass           2 wins from 5 races       GROUP B
013        Dakota          Roschelle       9          Hay and grass   7 wins from 10 races      GROUP D
014        Cisco           Nuwan           10         Grass           8 wins from 15 races      GROUP D
015        Spirit          Thomas          4          Hay             Champion-Mini Melbourne Cup GROUP D
016        Dusty           Rosie           6          hay             5 wins from 10 races      GROUP C
017        Sunshine        Jack            7          carrots and hay 9 wins from 14 races      GROUP C
018        Chief           Jill            6          grains          3 wins from 8 races       GROUP C
019        Night Rider     Joseph          5          Apples and grains 8 wins from 12 races      GROUP C
020        Hay Fever       Anton           5          Hay and grass   2 wins from 5 races       GROUP C
```

Flowchart:

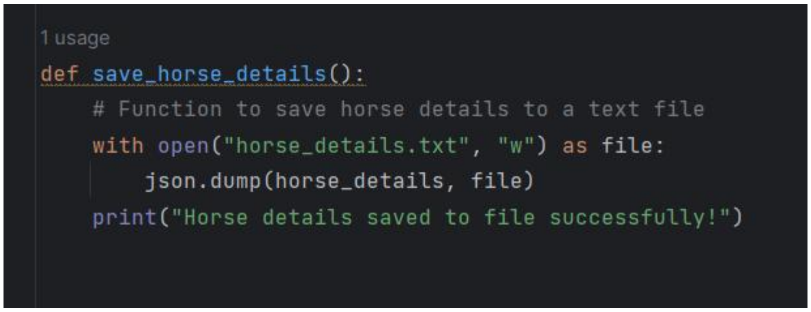

Start
Sort horse details by horse_id
Display a table with headers for each horse attribute
Display details for each horse in the sorted list
End

5. Function: save_horse_details()

Description:

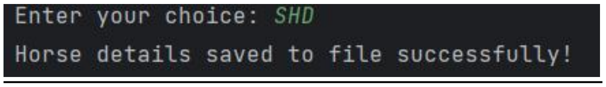
This function saves the current list of horse details to a text file named "horse_details.txt." The information is stored in JSON format, providing a convenient method for later retrieval.

Code:



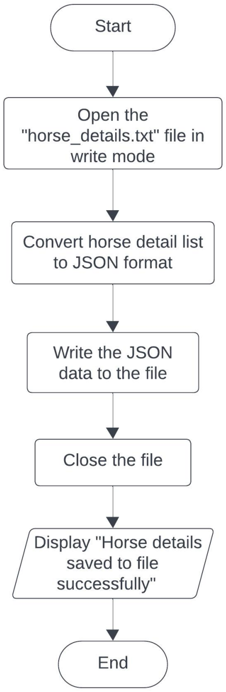
```
1 usage
def save_horse_details():
    # Function to save horse details to a text file
    with open("horse_details.txt", "w") as file:
        json.dump(horse_details, file)
    print("Horse details saved to file successfully!")
```

Result:



```
Enter your choice: SHD
Horse details saved to file successfully!
```

Flowchart:

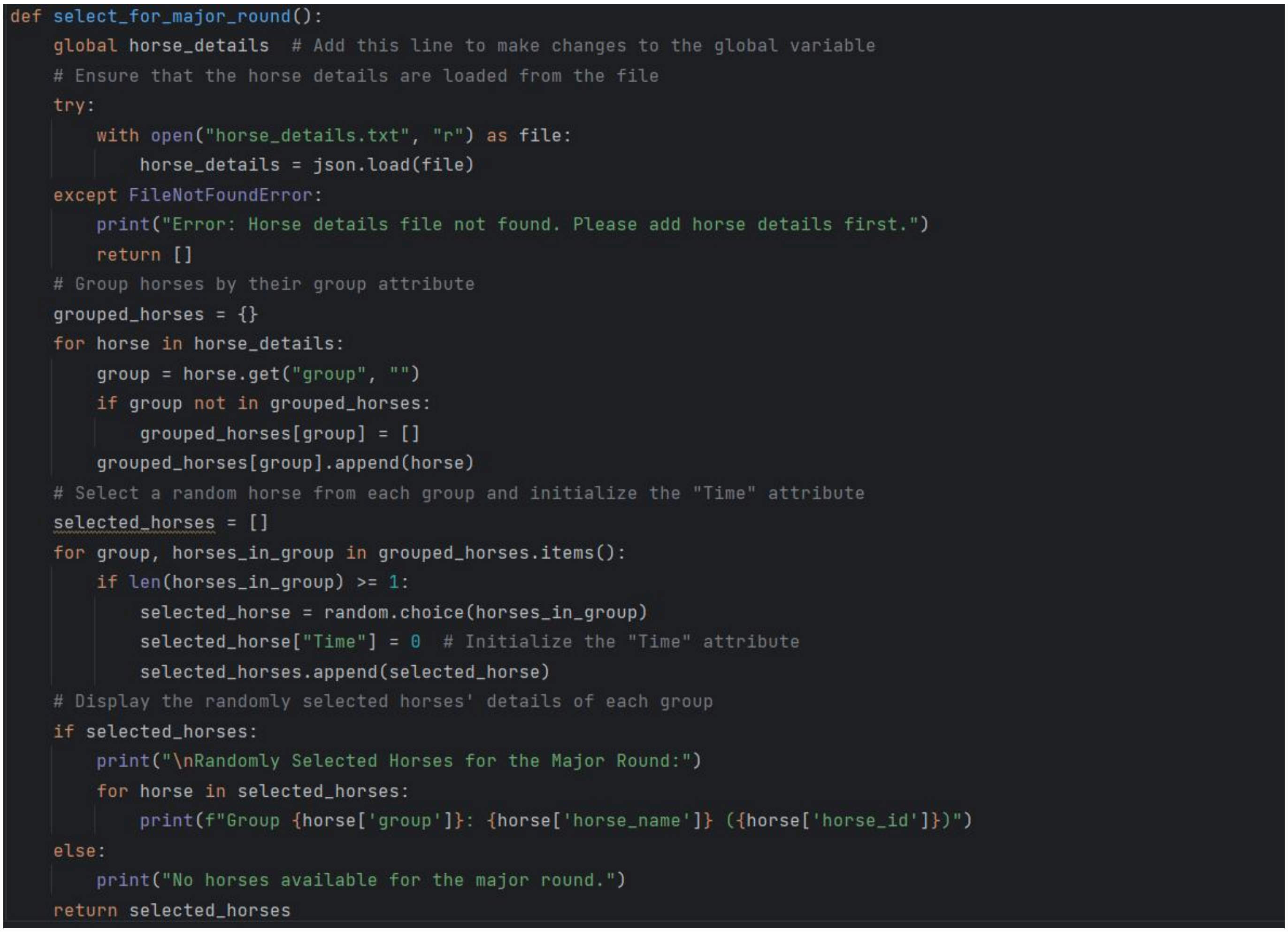

Start
Open the "horse_details.txt" file in write mode
Convert horse detail list to JSON format
Write the JSON data to the file
Close the file
Display "Horse details saved to file successfully"
End

6. Function: select_for_major_round()

Description:

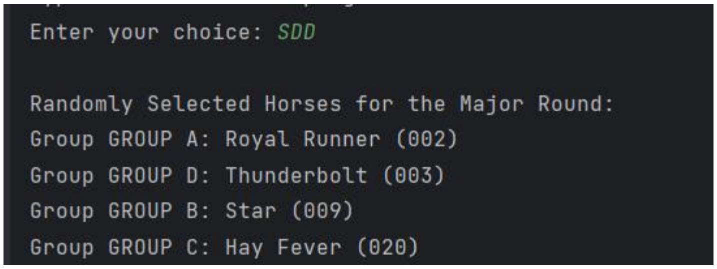
This function selects horses randomly for the major round of the race. It reads horse details from a file, groups them by their respective groups, and randomly selects one horse from each group. The selected horses are then initialized with a "Time" attribute, representing their race time.

Code:



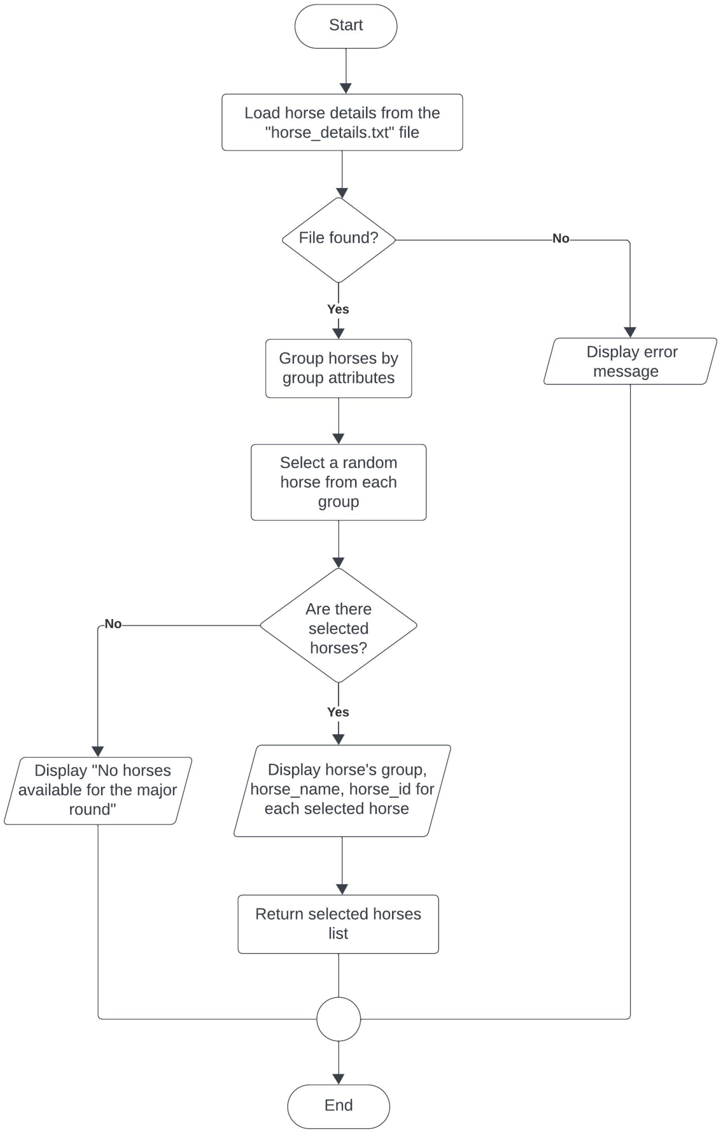
```
def select_for_major_round():
    global horse_details  # Add this line to make changes to the global variable
    # Ensure that the horse details are loaded from the file
    try:
        with open("horse_details.txt", "r") as file:
            horse_details = json.load(file)
    except FileNotFoundError:
        print("Error: Horse details file not found. Please add horse details first.")
        return []
    # Group horses by their group attribute
    grouped_horses = {}
    for horse in horse_details:
        group = horse.get("group", "")
        if group not in grouped_horses:
            grouped_horses[group] = []
        grouped_horses[group].append(horse)
    # Select a random horse from each group and initialize the "Time" attribute
    selected_horses = []
    for group, horses_in_group in grouped_horses.items():
        if len(horses_in_group) >= 1:
            selected_horse = random.choice(horses_in_group)
            selected_horse["Time"] = 0  # Initialize the "Time" attribute
            selected_horses.append(selected_horse)
    # Display the randomly selected horses' details of each group
    if selected_horses:
        print("\nRandomly Selected Horses for the Major Round:")
        for horse in selected_horses:
            print(f"Group {horse['group']}: {horse['horse_name']} ({horse['horse_id']})")
    else:
        print("No horses available for the major round.")
    return selected_horses
```

Result:



```
Enter your choice: SDD

Randomly Selected Horses for the Major Round:
Group GROUP A: Royal Runner (002)
Group GROUP D: Thunderbolt (003)
Group GROUP B: Star (009)
Group GROUP C: Hay Fever (020)
```

Flowchart:

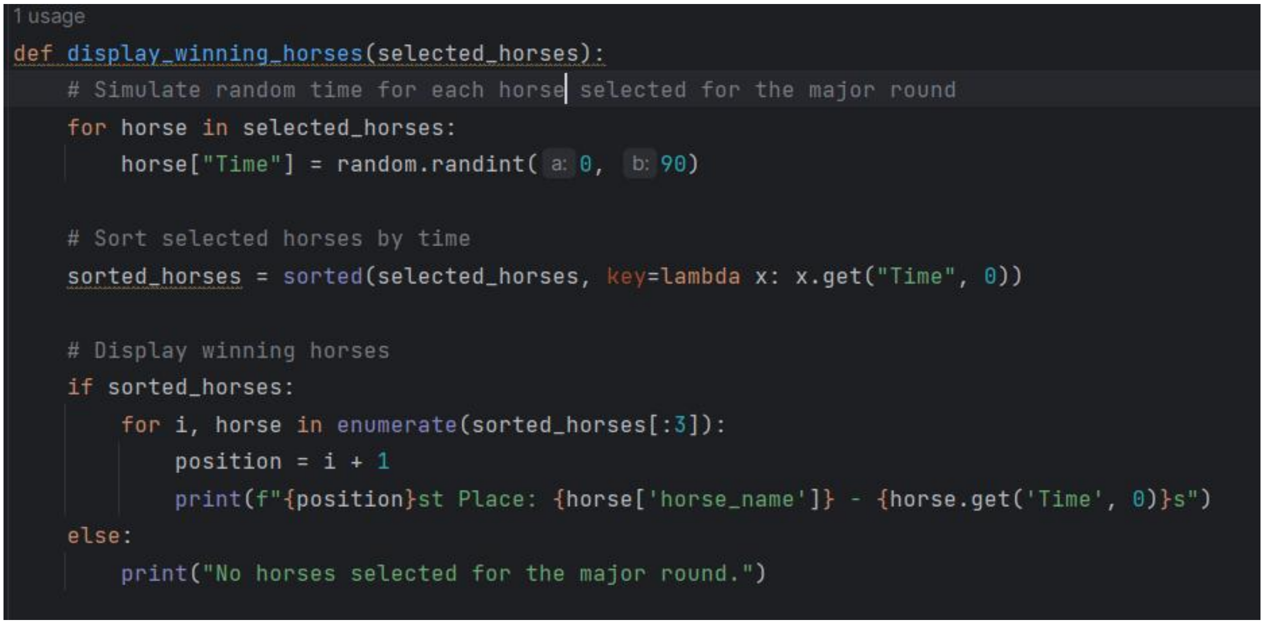

Start
Load horse details from the "horse_details.txt" file
File found?
No
Yes
Group horses by group attributes
Display error message
Select a random horse from each group
Are there selected horses?
No
Yes
Display "No horses available for the major round"
Display horse's group, horse_name, horse_id for each selected horse
Return selected horses list
End

7. Function: display_winning_horses(selected_horses)

Description:

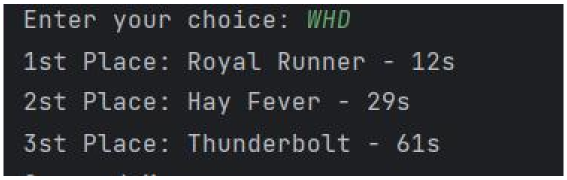
This function simulates random race times for the selected horses in the major round and displays the top three winning horses. The horses are sorted based on their race times, and the results are presented with their names and respective timings.

Code:



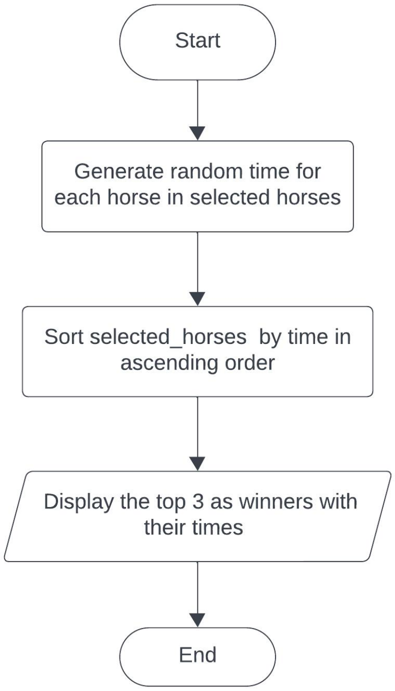
```
1 usage
def display_winning_horses(selected_horses):
    # Simulate random time for each horse selected for the major round
    for horse in selected_horses:
        horse["Time"] = random.randint( a: 0,  b: 90)

    # Sort selected horses by time
    sorted_horses = sorted(selected_horses, key=lambda x: x.get("Time", 0))

    # Display winning horses
    if sorted_horses:
        for i, horse in enumerate(sorted_horses[:3]):
            position = i + 1
            print(f"{position}st Place: {horse['horse_name']} - {horse.get('Time', 0)}s")
    else:
        print("No horses selected for the major round.")
```

Result:



```
Enter your choice: WHD
1st Place: Royal Runner - 12s
2st Place: Hay Fever - 29s
3st Place: Thunderbolt - 61s
```

Flowchart:

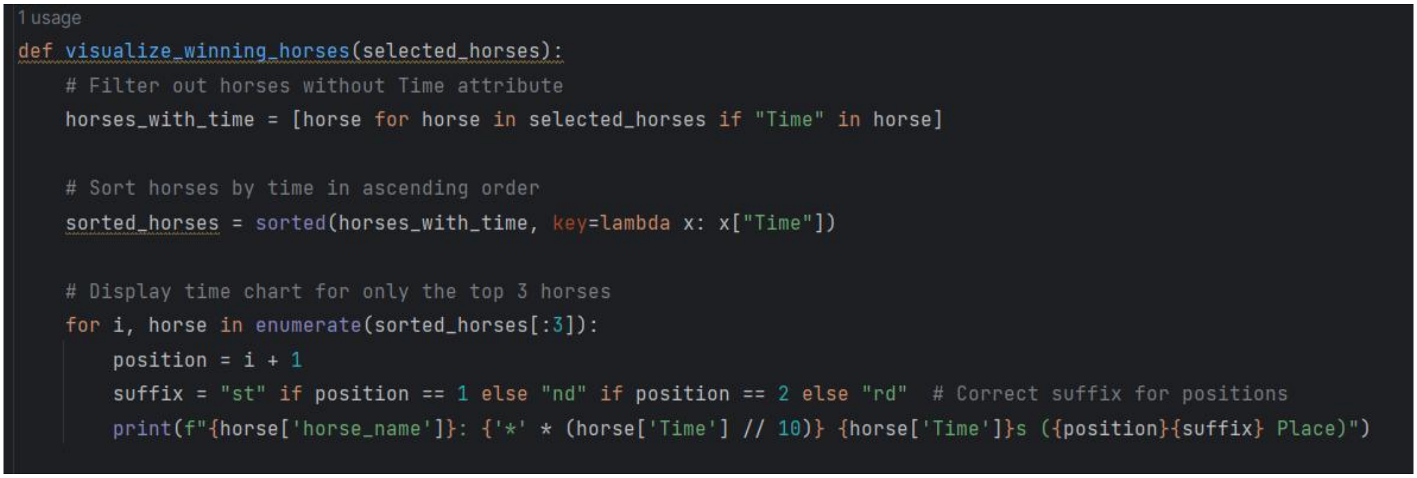

Start
Generate random time for each horse in selected horses
Sort selected_horses by time in ascending order
Display the top 3 as winners with their times
End

8. Function: visualize_winning_horses(selected_horses)

Description:

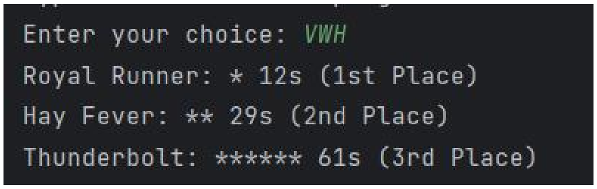
This function visualizes the race times of the winning horses by displaying a time chart. The chart consists of asterisks representing each 10 seconds of race time, providing a graphical representation of the horses' performance.

Code:



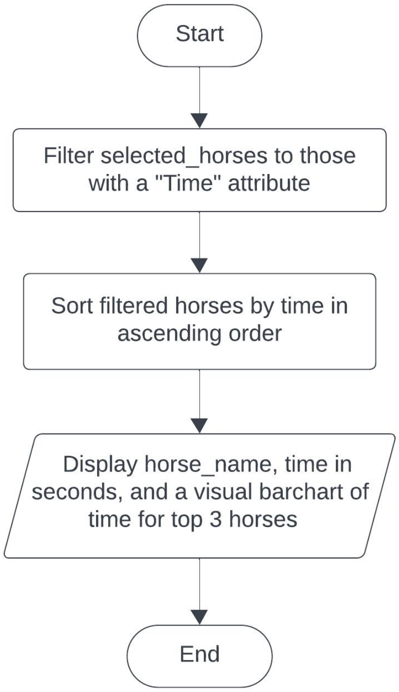
```
1 usage
def visualize_winning_horses(selected_horses):
    # Filter out horses without Time attribute
    horses_with_time = [horse for horse in selected_horses if "Time" in horse]

    # Sort horses by time in ascending order
    sorted_horses = sorted(horses_with_time, key=lambda x: x["Time"])

    # Display time chart for only the top 3 horses
    for i, horse in enumerate(sorted_horses[:3]):
        position = i + 1
        suffix = "st" if position == 1 else "nd" if position == 2 else "rd"  # Correct suffix for positions
        print(f"{horse['horse_name']}: {'*' * (horse['Time'] // 10)} {horse['Time']}s ({position}{suffix} Place)")
```

Result:



```
Enter your choice: VWH
Royal Runner: * 12s (1st Place)
Hay Fever: ** 29s (2nd Place)
Thunderbolt: ****** 61s (3rd Place)
```

Flowchart:

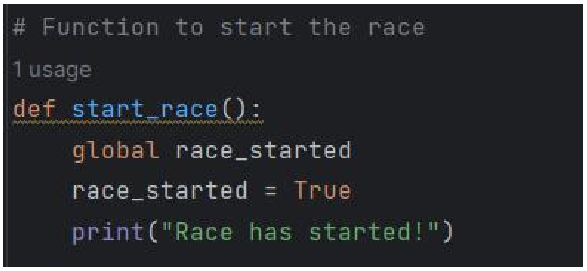

Start
Filter selected_horses to those with a "Time" attribute
Sort filtered horses by time in ascending order
Display horse_name, time in seconds, and a visual barchart of time for top 3 horses
End

9. Function: start_race()

Description:

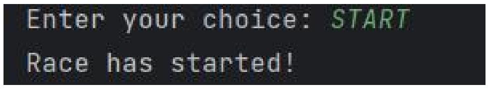
This function initiates the horse race by setting the race_started variable to True. Once started, certain functionalities, such as adding, deleting and updating horse details, are restricted to maintain the integrity of the race.

Code:



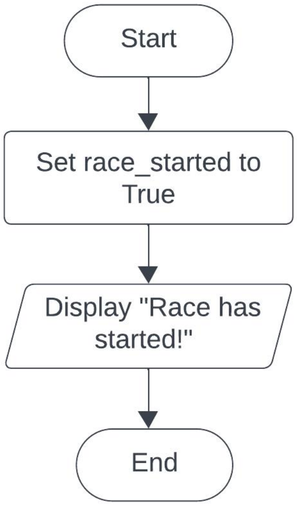
```
# Function to start the race
1 usage
def start_race():
    global race_started
    race_started = True
    print("Race has started!")
```

Result:



```
Enter your choice: START
Race has started!
```

Flowchart:



Complete Python code

```
import random

import json


horse_details = []

selected_horses = []

sorted_horses = []

race_started = False  # Variable to track whether the race has started


def adding_horse_details():

    # Function to add horse details

    if not race_started:

        horse_id =input("Enter horse ID: ")

        # Check if the horse with the same ID already exists

        existing_horse = next((horse for horse in horse_details if horse["horse_id"] == horse_id), None)

        if existing_horse:

            print(f"Horse with ID {horse_id} already exists:")

            print(existing_horse)

        else:

            horse_name = input("Enter horse name: ")

            jockey_name = input("Enter Jockey name: ")

            age = input("Enter horse age: ")

            breed = input("Enter horse breed: ")

            race_record = input("Enter horse race record: ")

            group = input("Enter horse group: ").upper()


            horse = {"horse_id":horse_id,"horse_name": horse_name,"jockey_name":jockey_name, "age": age, "breed": breed,"race_record":race_record, "group":group}

            horse_details.append(horse)

            print("Horse details added successfully!")


    else:

        print("Cannot add horse details after the race has started.")


def deleting_horse_details():

    # Function to delete horse details based on horse_id

    if not race_started:

        horse_id = input("Enter horse id to delete horse details: ")

        for horse in horse_details:
            if horse["horse_id"] == horse_id:
                horse_details.remove(horse)
                print("Horse details deleted successfully!")
                return
        print("Horse not found.")
    else:
        print("Cannot delete horse details after the race has started.")


def update_horse_details():
    # Function to update horse details based on horse_id
    if not race_started:
        horse_id= input("Enter horse id to update: ")
        for horse in horse_details:
            if horse["horse_id"] == horse_id:
                horse["horse_name"] = input("Enter new horse name: ")
                horse["jockey_name"] = input("Enter new jockey name: ")
                horse["age"] = input("Enter new horse age: ")
                horse["breed"] = input("Enter new horse breed: ")
                horse["race_record"] = input("Enter new race record: ")
                horse["group"] = input ("Enter new horse group: ").upper()
                print("Horse details updated successfully!")
                return
        print("Horse not found.")
    else:
        print("Cannot update horse details after the race has started.")


def view_horses_details():
    # Function to view horse details sorted by Horse ID
    sorted_horses = sorted(horse_details, key=lambda x: x.get("horse_id", 0))
    print("\nHorse Details Table (Sorted by Horse ID):")
    print("{:<10} {:<15} {:<15} {:<10} {:<15} {:<25} {:<10}".format(
        "horse_id", "horse_name", "jockey_name", "age", "breed", "race_record", "group"))
    for horse in sorted_horses:
        print("{:<10} {:<15} {:<15} {:<10} {:<15} {:<25} {:<10}".format(
            horse.get("horse_id", ""),
            horse.get("horse_name", ""),
            horse.get("jockey_name", ""),
            horse.get("age", ""),
            horse.get("breed", ""),
            horse.get("race_record", ""),
            horse.get("group", "")
        ))

def save_horse_details():
    # Function to save horse details to a text file
    with open("horse_details.txt", "w") as file:
        json.dump(horse_details, file)
    print("Horse details saved to file successfully!")



def select_for_major_round():
    global horse_details  # Add this line to make changes to the global variable
    # Ensure that the horse details are loaded from the file
    try:
        with open("horse_details.txt", "r") as file:
            horse_details = json.load(file)
    except FileNotFoundError:
        print("Error: Horse details file not found. Please add horse details first.")
        return []

    # Group horses by their group attribute
    grouped_horses = {}
    for horse in horse_details:
        group = horse.get("group", "")
        if group not in grouped_horses:
            grouped_horses[group] = []
        grouped_horses[group].append(horse)

    # Select a random horse from each group and initialize the "Time" attribute
    selected_horses = []
    for group, horses_in_group in grouped_horses.items():
        if len(horses_in_group) >= 1:
            selected_horse = random.choice(horses_in_group)
            selected_horse["Time"] = 0  # Initialize the "Time" attribute
            selected_horses.append(selected_horse)
    # Display the randomly selected horses' details of each group
    if selected_horses:
        print("\nRandomly Selected Horses for the Major Round:")
        for horse in selected_horses:
            print(f"Group {horse['group']}: {horse['horse_name']} ({horse['horse_id']})")
    else:
        print("No horses available for the major round.")
    return selected_horses

def display_winning_horses(selected_horses):
    # Simulate random time for each horse selected for the major round
    for horse in selected_horses:
        horse["Time"] = random.randint(0, 90)

    # Sort selected horses by time
    sorted_horses = sorted(selected_horses, key=lambda x: x.get("Time", 0))

    # Display winning horses
    if sorted_horses:
        for i, horse in enumerate(sorted_horses[:3]):
            position = i + 1
            print(f"{position}st Place: {horse['horse_name']} - {horse.get('Time', 0)}s")
    else:
        print("No horses selected for the major round.")

def visualize_winning_horses(selected_horses):
    # Filter out horses without Time attribute
    horses_with_time = [horse for horse in selected_horses if "Time" in horse]

    # Sort horses by time in ascending order
    sorted_horses = sorted(horses_with_time, key=lambda x: x["Time"])

    # Display time chart for only the top 3 horses
    for i, horse in enumerate(sorted_horses[:3]):
        position = i + 1
        suffix = "st" if position == 1 else "nd" if position == 2 else "rd"  # Correct suffix for positions
        print(f"{horse['horse_name']}: {'*' * (horse['Time'] // 10)} {horse['Time']}s ({position}{suffix} Place)")
```

```
# Function to start the race
def start_race():
    global race_started
    race_started = True
    print("Race has started!")

# Main program loop
while True:

    print("Command Menu:")
    print("Type AHD for adding horse details.")
    print("Type UHD for updating horse details.")
    print("Type DHD for deleting horse details.")
    print("Type VHD for viewing the registered horses’ details table.")
    print("Type SHD for saving the horse details to the text file.")
    print("Type SDD for selecting four horses randomly for the major round.")
    print("Type WHD for displaying the Winning horses’ details.")
    print("Type VWH for Visualizing the time of the winning horses.")
    print("Type START to start the race.")
    print("Type ESC to exit the program.")

    user_input = input("Enter your choice: ")

    if user_input == "AHD":
        adding_horse_details()
    elif user_input == "DHD":
        deleting_horse_details()
    elif user_input == "UHD":
        update_horse_details()
    elif user_input == "VHD":
        view_horses_details()
    elif user_input == "SHD":
        save_horse_details()
    elif user_input == "SDD":
        selected_horses=select_for_major_round()
    elif user_input == "WHD":
        display_winning_horses(selected_horses)
    elif user_input == "VWH":
```

```
    visualize_winning_horses(selected_horses)
elif user_input == "START":
    start_race()
elif user_input == "ESC":
    print("Exiting the program!")
    break
else:
    print("Invalid input. Please enter a valid command.")
```