**日報**

2023年08月08日

氏名：小松大二

|  |
| --- |
| ■目標  ・Problem12に取り組む  ＊燃料切れを起こしてレースが中断された場合は、順位判定を行わないように修正する  ＊carクラスを実装しレースに参加させる |
| ■作業内容  ・Problem12に取り組む  ＊燃料切れを起こしてレースが中断された場合は、順位判定を行わないように修正する  →完了  ＊carクラスを実装しレースに参加させる  →作業途中 |
| * 習得したこと   ・レースの中断処理を指示するif文は、条件が満たされた時のみ動くものなのでfor内に書く必要はないことを理解し、forの外に書く修正を行うことができる  ・returnを書くことで、メソッド内で処理が完了したり、特定の条件が満たされたりした場合に、メソッドの実行を終了し、それ以降のコードが実行されなくなることを理解できる  ・レースの中断処理が行われた場合、順位判定を行うjudgeメソッドが呼び出されないように、レースの中断処理にreturnを書くことで、それ以降のメソッドが実行されないようにし、順位判定を行わないようにすることができる |
| * 感想   今日は、Problem12の、燃料が切れてレースが中断される場合は、順位判定を行わないように修正する作業から入りました。日報のコメントで指摘があったように、「全てのボートの燃料が切れたのでレースを中断します」という部分の処理は、if文の条件が満たされた時のみ動くもので、for内に書く必要はないので、まずはforの外に書く修正を行いました。しかし、それだけではプログラムは順位判定を行ってしまい、どうしたらいいか分からず頭を抱えました。燃料切れを起こしてレースが中断された際には、順位判定を行っているjudgeメソッドを実行されないようにすればいいと考えたのですが、それをどのように書けば実装できるのか分からず、苦戦しました。if文をjudgeメソッドが実行されるところに書いてみるなどしたのですが、それでもうまくいきませんでした。そのため、考え方を変えて、燃料切れで中断された場合はそれ以降に書いてある処理が実行されないようにすればいいと考え、処理にreturnを追加しました。そうすることで、普通にレースが終了したらjudgeメソッドが実行され、レースが中断された場合はjudgeメソッドが実行されないように動かすことができました。結果は同じでも、考え方を変えるだけで、とてもシンプルに問題を解決することができたので、柔軟にアプローチ方法を考えることをこれからも意識していきたいです。  次にcarクラスを実装してレースに参加させる問題に取り組んだのですが、新しくクラスやインターフェースを作成しようとしたら、carクラスもTireクラスもどちらも既に存在していました。この問題は今あるクラスを削除して、また一から作って作業を進める問題なのか、それとも既存のcarクラス、Tireクラスを使うのか、どちらで進めていけばいいでしょうか。今日はとりあえず、Problem12の1問目を解いたときと同じ要領で、新しく車を追加したところまで進めました。 |