2.1. ロケットクラス（概念と実体）--燃料ロケット--

Rocketというクラスを考えてみましょう。Rocketクラスのインスタンスとして燃料ロケット、原子力ロケット、ワープロケット等が考えられますが、まずは基本の燃料ロケットの仕様を検討し実体化してみます。

Rocketクラスはメンバーフィールドとして燃料とスピード、メンバーメソッドとしてコンストラクタ()と加速()、燃料値の取得()、スピード値の取得()を定義します。初期燃料の値はオーバーロードされたコンストラクタで設定します。

以下、個々のロケットの仕様を提示します。

【ロケットの仕様】

(1)燃料ロケット
・燃料の初期値をコンストラクタに引数で設定します。
・加速すると燃料は-2消費され、速度は+2増加します。
・燃料が2より小さくなると加速できません。

(2)原子力ロケット
・燃料ロケットを継承します。
・燃料は必要としませんのでコンストラクタに引数はありません。
・速度は１回の加速で+10の増分値で100まで加速可能です。

(3)ワープロケット
・燃料ロケットを継承します。
・燃料は必要としませんのでコンストラクタに引数はありません。
・速度は１回の加速で+100の増分値で無限大まで加速可能です。

【実習１】
以下のクラス図に基づきロケット仕様の(1)について、クラスを実装し実行結果となるようなプログラムを作成してください。
なお、実行メソッドは提示します。

（仕様(1)クラス図）

クラス名
Rocket

---
fuel:int
speed:int

+ Rocket():
+ Rocket(firstFuel:int):
+ getFuel():int
+ getSpeed():int
+ kasoku():void

+ main(String args):staic void

-:private
+:public
#:protected

（実行メソッド）

```
public static void main(String[] args){
    System.out.println("燃料ロケットをメモリ上につくります。");
    System.out.print("燃料（整数）を入力してください：");
    int f = new java.util.Scanner(System.in).nextInt();
    Rocket roc=new Rocket(f);
    roc.kasoku();
    roc.kasoku();
    roc.kasoku();
}
```

（実行結果）

```
燃料ロケットをメモリ上につくります。
燃料（整数）を入力してください：5
【燃料ロケット】
燃料で加速します！
速度が上がりました
現在の燃料：3
現在の速度：2
【燃料ロケット】
燃料で加速します！
速度が上がりました
現在の燃料：1
現在の速度：4
【燃料ロケット】
燃料不足で加速できません！
現在の燃料：1
現在の速度：4
```