Gadget. Java

```
public String getModel() {
  return model;
}
```

This is a getter method for the private string variable ‘model’

```
public double getPrice() {
  return price;
}
```

This is a getter method for the private double variable ‘price’

```
public int getWeight() {
  return weight;
}
```

This is a getter method for the private int variable ‘weight’

```
public String getSize() {
  return size;
}
```

This is a getter method for the private string variable ‘size’

Mobile. Java

```
public int getCreditRemaining() {
  return creditRemaining;
}
```

This method returns an int value which is ‘creditRemaining’

```
public String addCredit(int amount) {

  if (amount <= 0) {
    // Invalid amount entered.
    System.out.println("Error! Invalid amount entered. Please enter a value greater than zero.");
    return "Error! Invalid amount entered. Please enter a value greater than zero.";
  }
  else {
    this.creditRemaining += amount;
    System.out.println("Success! " + amount + " is added to your credit.");
    return "Success! " + amount + " is added to your credit.";
```

```
  }
}
```

This method adds credit(int). It first checks if the credit that was inputted was less or equal to zero if that's the case an error message is printed if its more than zero then the inputted int is added to 'creditRemaining' and a message saying success the (int amount) has been added.

```
public String call(String phoneNumber, int duration) {
  if (this.creditRemaining < duration) {
    // User does not have enough balance for this call.
    System.out.println("Error! Insufficient credits remaining. Cannot make this call. "
        + "Remaining Credit: " + this.creditRemaining);


    return "Error! Insufficient credits remaining. Cannot make this call. "
        + "Remaining Credit: " + this.creditRemaining;
  }
  else {
    // User has enough credits for this call.
    this.creditRemaining -= duration;
    System.out.println("Success! Phone Number: " + phoneNumber
        + ", Duration: " + duration
        + ", Remaining Credit: " + this.creditRemaining);


    return "Success! Phone Number: " + phoneNumber
        + ", Duration: " + duration
        + ", Remaining Credit: " + this.creditRemaining;
  }
```

This method stimulates a call and deducts the cost of the call from the users remaining credit. If the users credit balance is too low to make the call an error message will print. If the credit balance is enough the call will be made and will deduct the credit from the balance and will print out a message saying the remaining credit left in the balance

Mp3.Java

```
public String downloadMusic(int memoryRequired) {
  if (this.memoryAvailable < memoryRequired) {
    // User does not have enough memory for this download.
    System.out.println("Error! Insufficient memory remaining. Cannot download. "
        + "Available Memory: " + this.memoryAvailable);


    return "Error! Insufficient memory remaining. Cannot download. "
        + "Available Memory: " + this.memoryAvailable;
  }
  else {
    // User has enough memory for this download.
    this.memoryAvailable -= memoryRequired;
    System.out.println("Success! Available Memory" + this.memoryAvailable);
```

```
    return "Success! Available Memory: " + this.memoryAvailable;
  }
}
```

This method stimulates a download of a song. If the memory isn't enough to download the song an error message will get printed. If there is enough memory for the song to get downloaded, then the memory will get deducted from the memory storage and a message will print the remaining memory.

```
public String deleteMusic(int memoryReleased) {
  this.memoryAvailable += memoryReleased;

  System.out.println("Success! Music deleted. Available Memory: " + this.memoryAvailable);
  return "Success! Music deleted. Available Memory: " + this.memoryAvailable;
}
```

This method stimulates deleting a song from the Mp3. When the music is deleted it will return the memory that the music used back to the memory storage and will print a message stating the new memory balance.