課題2について

学籍番号：（1830176）　　　　氏名（寺田縁）

1. 追加機能の説明（対象者、目的、効果など）

今回、追加機能として、県ごとに降水確率が表示される機能を実装した。機能実装のために、drk7.jp(<https://www.drk7.jp/weather/>)というサイトから、新たにXMLファイルを導入している。これは、気象庁が公開している天気予報情報を独自の手法でメタデータ化し、XML形式で配信しているサービスであり、6時間ごとに降水確率が記録されている。今回は、それらのデータから降水確率の平均値を算出し、一日の降水確率として表示している。降水確率の情報を追加することで、出かける際に、雨具を持っていくかどうかといった判断がつきやすくなると考える。

閲覧方法は簡単であり、地図上の県のイラストをクリックすると、天気情報や最高、最低気温に加えて、降水確率が表示される。ただし、今回参考としたXMLのファイルには佐渡が含まれていないため、佐渡をクリックすると、表示できないというメッセージが表示される仕様にしている。

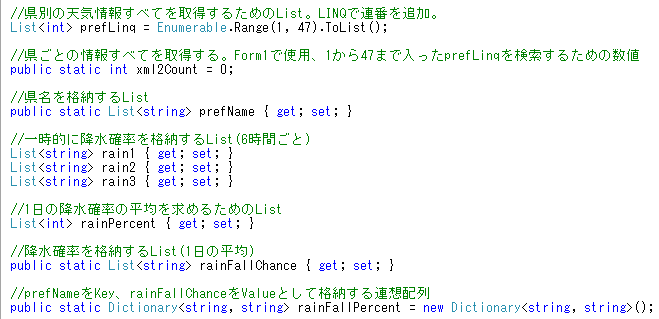
1. プログラム内容の説明（連想配列、ラムダ式、LINQ、XML、非同期／並列化の説明）

今回は、指定された仕組みをすべて使用した。

それらの仕組みをどのように使用したか、及びどのような流れで降水確率の情報を表示しているかについて説明していく。

まず、降水確率を取得するためのクラスとして、RainFallを実装している。その後、以下のリストおよび連想配列を実装している。

prefLinqはLINQ機能を利用し、1から47の連番を追加している。このリストは、XMLのデータを取得する際に使用する。加えて、prefLinq内の値を取得するための変数として、xml2Countを実装している。そして、県名を格納するためのリストとして、prefNameを実装している。降水確率は、まず、6時間ごとの降水確率を格納するためのリストとして、rain1～rain3を実装している。0時～6時は、必ず0％として記録されているため、今回は除外している。次に、rain1～rain3から得られた降水確率から、平均値を求めるために、rainPercentを実装している。最後に、rainPercentの数値をテキストデータとして記録するrainFallChanceを実装し、prefNameをKey、rainFallChanceをValueとして格納する連想配列、rainFallPercentを実装している。これらの配列は後に使用する。



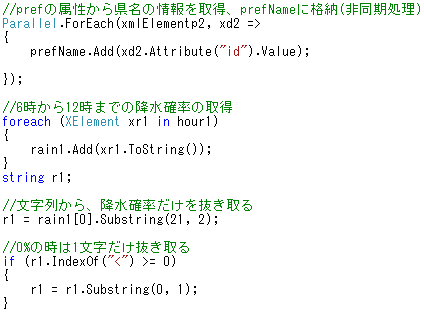
次に、XMLファイルを取得していく。これは、すでに取得しているYahooのXMLファイルの取得方法を参考にしている。

今回、取得したXMLファイルのURLは、共通部分＋識別番号(01〜47)＋.xmlという構成であり、共通部分と.xmlは手入力で入力し、県ごとに異なる識別番号はprefLinqから入力される仕組みである。xml2Countで検索することで、1から47の数字をURLに代入し、全ての県のXMLファイルを取得している。また、一桁の識別番号は01といった仕様であるため、1から9の数字には0を文字として付け足している。ファイル取得後は、県名、降水確率を取り出している。

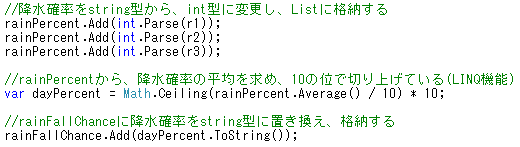


降水確率は、LINQを利用し、区切られている時間ごとに要素を検索することで、それぞれの時間ごとの降水確率を取得している。

　次に取得したデータを、実装しているリストに格納していく。

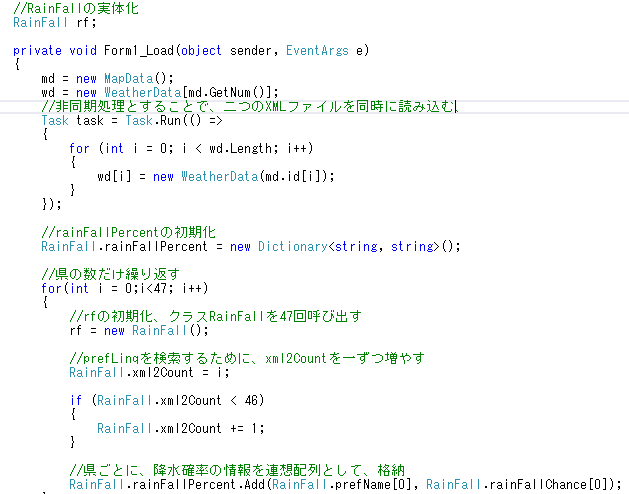


県名はprefNameに格納している。加えて、処理を非同期処理で行っている。降水確率は区切られている時間ごとにrain1～rain3に一時的に格納している。この段階では、まだ余計な文字列が含まれているため、降水確率の文字列だけを抜き取っている。さらにif文を用いることで、抜き出された文字列が0だった際に、余計な文字列が含まれないようにしている。



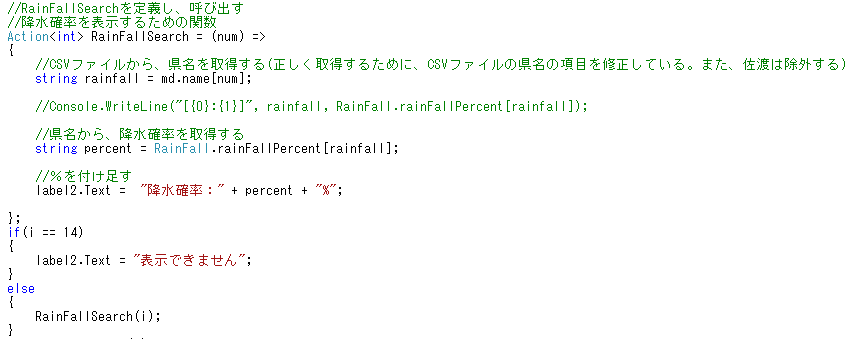
次に、抜き取った降水確率をstring型からint型に変更し、rainPercentに格納している。そして、LINQを用いることで、平均値を算出、10の位で切り上げている。算出された平均値は再び、string型に変更し、rainFallChanceに格納している。

以上が、RainFallクラスで記述したコードである。次に、Form1に新しく記述したコードについて説明する。



まず、RainFallクラスをrfとして実体化している。その後、連想配列rainFallPercentを初期化している。何度も呼び出すこととなるRainFallクラスで初期化してしまうと、県別の降水確率が記録されていかないので、Form1で一度だけ初期化される仕様にしている。rainFallPercentを初期化した後は、RainFallクラスを県の数、つまり47回呼び出し、全ての県のXMLファイルを取得している。このとき、xml2Countを一ずつ増やすことで、prefLinqから取得される値が変化し、URLが変更されていく。そして、一つの県の情報が取得されるたびに、rainFallPercentに県名がKey、降水確率がValueとして格納されていき、全ての県の降水確率が格納される。また、新たにXMLファイルを追加したことで、処理速度が低下してしまう恐れがあるため、Taskを用いることでYahooのXMLファイルを読み込む処理と、降水確率のXMLファイルを読み込む処理を並列で行うようにしている。

次に、rainFallPercentに格納した降水確率を、実際に表示させるための関数、RainFallSearchをラムダ式で実装している。



引数をnumとし、他の関数と同様に、クリックされた県の番号を引数として使用する。Valueに当たる降水確率は、Keyである県名から検索し、取得している。MapDataで実装されているname配列を利用し、引数numに該当する県名をstring型のrainfallに代入することでrainFallPercent内を検索、取得した降水確率をstring型のpercentに代入し、テキストデータとしてlabelに表示している。加えて、この関数はForm1がロードされた際にも引数を0として定義し、呼び出している。また、県名による降水確率の検索にあたって、CSVファイルと、今回取得したXMLファイルとの間で、県名の表記が食い違っていたため、CSVファイルの記述を修正している。

最後に、実装した関数、RainFallSearchを呼び出す。佐渡にあたる14が引数として返された際は、RainFallSearchは呼び出さず、labelの表記を、表示できませんに変更している。