

予約サービス向けの祝日休日自動計算ライブラリの開発

金政 拓実¹ 井手 伊織² 成 凱^{1,a)}

概要: 祝日・休日は予約サービスをはじめ、多くのビジネスにとって欠かせない情報である。祝日とは、法律により定められた休日であり、日本においては「国民の祝日」と称される。一方で、企業活動を休業する日については「休日」と呼ばれる。週末や祝日を休日とするのが一般的ではあるが、業種・企業や店舗・職場によって土日や祝日に完全休業しているとは限らない。例えば、飲食店やサービス業は、週末や祝日に集客が見込まれるため、休まずに営業する機会が多い。また、祝日も法律の改正や特例により祝日が変わることもありうる。祝日休日を正確に計算することは、予約サービスに必要不可欠である。本研究では、祝日休日自動計算ライブラリを開発し、年度（暦年、会計年度、事業年度、学校年度）、半期、4半期、月度、週間、日をそれぞれ時間単位として、祝日・休日の正しい計算を可能にする。また、祝日に関する法律の改正や、業種・企業・店舗の営業予定に合わせて設定ファイルによるカスタマイズ機能を開発し、汎用性の高いライブラリを実現する。

A PHP Library for Calculating Public Holidays and Business Days in Reservation Services

1. はじめに

祝日・休日は予約サービスをはじめ、多くのビジネスにとって欠かせない情報である。祝日とは、法律により定められた休日であり、日本においては「国民の祝日」と称される。一方で、企業活動を休業する日については「休日」と呼ばれる。週末や祝日を休日とするのが一般的ではあるが、業種・企業や店舗・職場によって土日や祝日に完全休業しているとは限らない。例えば、飲食店やサービス業は、週末や祝日に集客が見込まれるため、休まずに営業する機会が多い。また、祝日も法律の改正や特例により祝日が変わることもありうる。

祝日休日を正確に計算することは、予約サービスに必要不可欠である。予約サービスは、一般的に年度ごとに営業日・休業日を事前にきめて営業計画を立てる必要がある。営業日・休業日を決める際に曜日や祝日に関する情報が

重要である。祝日・休日を計算するためには、Yasumi[1] や JapaneseDate[2] といったライブラリやコンポーネントが存在するが、法律改正による祝日の追加変更や、営業日・休業日の設定等はプログラムの変更や追加の実装が必要となり、汎用性に欠けている。

本研究では、祝日休日自動計算ライブラリを開発し、年度（暦年、会計年度、事業年度、学校年度）、半期、4半期、月度、週間、日をそれぞれ時間単位として、祝日・休日の正しい計算を可能にする。また、祝日に関する法律の改正や、業種・企業・店舗の営業予定に合わせて設定ファイルによるカスタマイズ機能を開発し、汎用性の高いライブラリを実現する。

2. 祝日休日計算に関するドメイン知識

祝日休日計算に関連するドメイン知識をまとめる。祝日休日は学校や、行政機関や企業の年度範囲の活動計画に係るものであり、年度を理解することが重要である。年度は、カレンダー上の一年間と異なる場合が多い。カレンダー上の一年間は、暦年と呼ばれ、暦法に基づいて制定されたものである。また、企業などの活動は、月、週単位に行われることが多く、「毎月第3木曜に休業日とする」等の使いがよく見られる。図1は、これらの用語とその関係性を示す。

¹ 九州産業大学理工学部情報科学科
Department of Information Science
Kyushu Sangyo University,
2-3-1, Higashi-ku, Fukuoka, 813-8503, Japan

² 九州産業大学大学院情報科学研究科
Graduate School of Information Science
Kyushu Sangyo University

a) chengk@is.kyusan-u.ac.jp

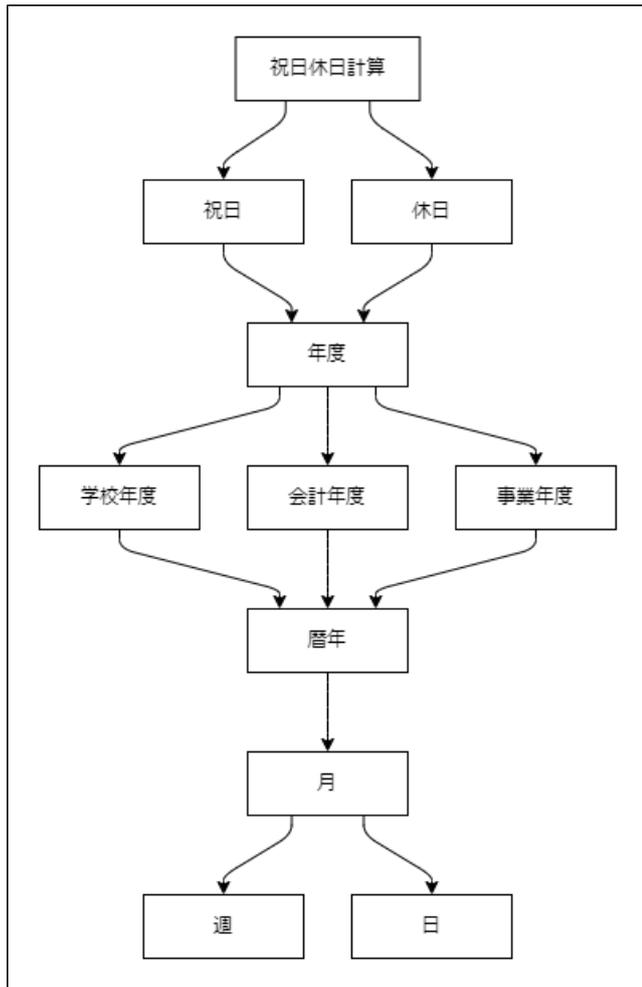


図 1 祝日休日計算に関するドメイン知識

2.1 暦年

暦（こよみ、れき）とは、時間の流れを年・月・週・日といった単位に当てはめて数えるように体系付けたもの、また、その構成の方法論（暦法）や、それを記載した暦書・暦表（日本のいわゆる「カレンダー」）を指す。さらに、そこで配当された各日ごとに、月齢、天体の出没（日の出・日の入り・月の出・月の入り）の時刻、潮汐（干満）の時刻などの予測値を記したり、曜日、行事、吉凶（暦注）を記したものをも含める。表 1 は暦に関する用語をまとめている。

日本では、明治維新（1868 年）以前の日本で使われていた暦を「旧暦（陰暦）」と呼び、は明治 6 年（1873 年）から使われている暦を「新暦（太陽暦）」と呼ばれる。21 世紀現在ではほとんどの国家でグレゴリオ暦が使用されている状況である。

グレゴリオ暦は、太陽の動きをもとにして作られているため、「太陽暦」と呼ばれる。平年では 1 年を 365 日とし、400 年間に 100 回ではなく 97 回の閏年を置いてその年を 366 日とする。400 年間における 1 年の平均日数を $365 + (97/400)$ 日 = 365.2425 日（365 日 5 時間 49 分 12 秒）とする。この平均日数 365.2425 日は、実際に観測で求め

表 1 暦に関する用語

用語	説明
暦（こよみ）	日を記録するもの
暦法	暦の方法論（新暦、旧暦）
暦表	暦による日付の並びを表形式等で表示したもの（通称「カレンダー」）
天体暦	天象の予報・天体の軌道を記述するもの
航海暦	航海用に一年間の天象・天体の視位置を記述したもの
生活暦	日ごとに天象に加えて行事や占いや曜日などを総合して記述したもの
グレゴリオ暦	ローマ教皇グレゴリウス 13 世が 1582 年に制定した太陽暦。現在世界中で使用されている。
旧暦	明治維新（1868 年）以前の日本で使われていた暦（陰暦）
新暦	明治 6 年（1873 年）から使われている太陽暦

られる平均太陽年（回帰年）の 365.242189572 日（2013 年中央値）に比べて約 26.821 秒長いだけであり、精度が高いと知られている。

暦年には、月、週、日等を用いて時間を表すことが一般的である。一年間を 12 か月、一週間を 7 日間としている。日本語では、曜日が循環する 7 日の組を週とし、各曜日に、日曜日・月曜日・火曜日・水曜日・木曜日・金曜日・土曜日の七曜を冠する。七曜（7 つの天体）が守護するとされる日を表す語句である。ある年の最初の週はその年の最初の木曜日を含む週である。これは、「ある年の 1 月 4 日を含む週がその年の最初の週である」という規則と同等である。ある年の最後の週は、次の年の最初の週の 1 つ前の週である。

2.2 年度について

ビジネスや行政サービスなどは、年度単位で計画を立てたりしている。年度は、暦年とは別に、特定の目的のために規定された 1 年間の区切り方であり、ビジネスや社会活動に広く影響を及ぼしている [3][4]。日本では多くの場合、グレゴリオ暦の 1 月 1 日以外を開始点とするものが狭義の「年度」とされ、暦年と区別する意味で「年度」と呼んでいる。ただし、暦策定の目的ではない特定の目的で計算などをするために「1 年」を定める必要がありそのための開始日を任意に定めており、たまたま 1 月 1 日でも良いのでそうしている場合も「年度」と呼ぶ。

年度は様々な決め方がある。法令上、「年度」の代表的な例として、「会計年度」と「事業年度」が存在する一方、「学校年度」、「生産年度」なども広く使われている。表 2 は各種年度の概要をまとめている。

学校年度は、学校で定められた、1 年間を単位とした修学期間。「学年」とも呼ばれるが、学年は年次の意味もあるので、混乱するかもしれない。学校年度は国や地域によって違う。日本、インド、パキスタン、ネパール、北朝鮮は、

表 2 各種年度の概要

年度	説明
暦年	暦の上で決めた 1 月から 12 月までの一年間
学校年度	学校で定められた 1 年間を単位とした修学期間
会計年度	国や法人等の収入および支出を整理分類し状況を明らかにするために始期日と終了日を定めた 1 年間
事業年度	企業の税金計算に必要な所得の計算期間
生産年度	農産物や加工品、工業製品に関する年度

4 月からの 1 年間を学校年度としている。一方、ヨーロッパ、アメリカ、トルコ、モンゴル、中国などは、9 月からの 1 年間を学校年度としている。世界的に見ると 9 月開始が圧倒的な多数派である。

会計年度は、国、地方自治体または法人の収入および支出を整理分類し、その状況を明らかにするために始期日と終了日を定めた 1 年間である。会計年度も国や地域によって異なる。

事業年度は、税金計算に必要な所得の計算期間である。会社では会計期間で計算した損益をもとに納付する税金を計算し申告書を作成する、つまり税務処理を行う必要がある。税金を計算する期間は会社法と法人税法で決められており、この期間を事業年度という。事業年度は、会社で自由に決められる期間に対して、会計年度は、法律や規制に基づいて設定される期間になる。

その他、農作物・加工品関連の年度として、砂糖年度、穀物年度、トウモロコシ年度などがあり、また、工業製品関連の年度として、農薬年度、肥料年度、冷凍年度などが使われている。

2.3 国民の祝日

日本の「国民の祝日」は、法令の改正や国レベルの大行事に伴う特例の祝日等もあり、時代とともに増え、その内容が変化してきた [5][6]。表 3 にはこれまで日本の「国民の祝日」の変遷をまとめている。明治期に制定した「休假日」を、大正期には「休日」と改称し、さらに戦後の立憲君主制（象徴天皇制）のもとでは「国民の祝日」に変えた。「国民の祝日」は、経済発展を成し遂げた先進国として、ゆとりができたことを示している。つまり、週休二日制の定着に加えて月曜日固定の「ハッピーマンデー制度」や「振替休日」・「国民の休日」まで新設され連休が多くなった。

国民の祝日はこれまでに皇室慶弔行事に伴う休日や、東京オリンピック開催等に伴い増減してきた。今後も法律改正や特例による新しい祝日が生まれるかもしれない。祝日休日計算ライブラリは、これらの状況を想定して汎用性の高い設計をしないとけない。

2.4 営業日・休業日

営業日・休業日は、企業などの営業計画によって決められるものであり、業界によってさまざまな呼び方がある。

表 3 「国民の祝日」の変遷

月	日	適用期間	祝日名	
1	1	1948～	元日	
	15	1948～1999	成人の日	
	第 2 月曜	2000～		
2	11	1966～	建国記念日	
	23	2019～	天皇誕生日 (令和天皇)	
	24	1989	大喪の礼	
3	春分日	1948～	春分の日	
4	10	1959	明仁親王の結婚の儀	
	29	1948～1988	天皇誕生日 (昭和天皇)	
	29	1989～2006	みどりの日	
	29	2007～	昭和の日	
5	1	2019	天皇即位の日	
	3	1948～	憲法記念日	
	4	2007～	みどりの日	
	5	1948～	こどもの日	
6	9	1993	徳仁親王の結婚の儀	
	20	1995～2002	海の日	
	22	2021		
	23	2020		
	7	第 3 月曜	2003～2019 2022～	スポーツの日
		23	2021	
8	24	2020	山の日	
	8	2021		
	10	2020		
9	11	2016～2019 2022～	敬老の日	
	15	1966～2002		
	第 3 月曜	2003～		
10	秋分日	1948～	秋分の日	
	10	1966～1999	体育の日	
	第 2 月曜	2000～2019		
	第 2 月曜	2022～	スポーツの日	
11	22	2019	即位礼正殿の儀	
	3	1948～	文化の日	
	12	1990	即位礼正殿の儀	
12	23	1948～現在	勤労感謝の日	
	23	1989～2018	天皇誕生日 (平成天皇)	

学校等では、授業日、休業日と呼ばれ、病院では、受診日・診察日・診療日、休診日、店舗や飲食店では、営業日、店休日・休館日などの呼び方を使っている。また、休業日は、非営業日、営業日外とも呼ばれる。そのほか、実営業日、1 営業日、1 営業日前、中営業日などの表現も使われる場合がある。

ビジネスで多用される営業日の使い方としては、「○営業日以内」や「○営業日後」が挙げられる。計算方法の基

本として、起算日の後に続く休業日は営業日にカウントされない。「翌営業日」は「起算日から計算して次の営業日」のこと。「前営業日」は「起算日より前の営業日」のこと。「週末や祝日はお休みとなりますので、前営業日までにご入金いただけますようお願いいたします。」といった利用例がある。休日や祝日の対応ができないため、土日や祝日前の営業日に来店や入金を求めることが一般的である。これを「前営業日」と呼び、特に休日の直前ではなく実際に営業している日（実営業日）を指す。「休前日」は主に週末の前日、通常は金曜を指すが、祝日の場合は曜日に関わらず、直前の日を意味する。

営業日・休業日の決め方は、業種・企業による違い。土日や祝日に完全休業しているとは限らない。例えば、美容院は、月曜か火曜に休業するケースが多く、不動産屋の場合、一般的な定休日は水曜日。飲食店やサービス業は、週末に集客が見込まれるため、月曜に休む傾向が見られる。これらの職場や店舗の場合、土日は休みでなく営業日になる一方、月曜や水曜は、平日でも休業日と見なされる。

一般的な週休パターンとしては、(1) 週休2日制：月曜日から金曜日を営業日とし、土日休み。多くのオフィスワークや企業で採用される。(2) 週休1日制：週に1日だけ休業日が設定される。曜日は不定の場合も。小売業や飲食業など。(3) 変則的な休み：週ごとに休みの日数が変わる。例えば、第1・3土曜日は営業、第2・4土曜日は休みなど。(4) グローバル企業：各国の祝日や文化に対応した営業日を設定する。日本の祝日に休まないが、現地法人の休日に合わせる場合もある。(6) お盆や特定地域の慣習：地域独自の休業日（例：お盆休み）を設ける企業もある。

3. 祝日自動計算

本節は、祝日自動計算の仕組みを説明する。祝日に限られる種類しかなく、種類別にルールを決めれば、毎年の祝日を、その年に適用できるルールに基づき、自動的に計算できる。

3.1 祝日の種類

種類別に祝日を計算する方法を考える。表4に示すように、祝日は日付指定、曜日指定、特別な計算、「振替休日」、「国民の休日」の5パターンに分けることができる。

日付指定の祝日は、そのまま日付が決まる。曜日指定の場合は、その年その月の曜日から、日付を計算する必要がある。次に、春分の日、秋分の日は、天文学の知識に基づき計算する必要があり、ルール化することができない。通常の祝日が決まった後、振替休日を求めて祝日に追加する。最後に、国民の休日を求めて祝日に追加する。

上記の5パターンは既存の祝日に関する法律にすべて対応できる。将来新しい法律ができて、これらのパターン以内ならば、問題なく対応できる。

表4 「国民の祝日」の種類

種別	説明
日付指定	日付が固定されている日（元日、建国記念の日、文化の日など）
曜日指定	「〇月の第〇月曜日」のように曜日が定められている日（成人の日、敬老の日など）
特別な計算	天文学上の言葉である「春分の日」と「秋分の日」
振替休日	「国民の祝日」が日曜日に当たるときは、その日後においてその日に最も近い「国民の祝日」でない日
国民の休日	「国民の祝日」でない日でその前日及び翌日が「国民の祝日」である日

3.2 祝日定義の仕組み

上記のパターンをルールすることで、祝日計算は、ルールベースで自動的に行うことができる。本小節では祝日のパターンをルール化するための仕組みについて説明する。

3.2.1 曜日を指定する方法

計算上の便利で曜日は0から順に数字を割り当てる必要がある。月曜日は1として、火曜は2、水曜は3のようにする。週の始まりが日曜日として日曜日に0を割り当てる。つまり

0: 日, 1: 月, 2: 火, 3: 水, 4: 木, 5: 金, 6: 土

祝日休日を定める際に、毎月「第2月曜」や「第2第4第4木曜」のように指定する必要がある。一つの曜日に対応して、このような曜日を指定するため、以下のような配列を用いる。

$$[w, n_1, n_2, \dots]$$

w は曜日を表す整数 ($w \in [0..6]$) であり、 n_1, n_2, \dots は第何の曜日を指定する。ただし、「第1水曜と第2の月曜」のように曜日が異なる場合は、別の配列を使って対応する必要がある。同時に指定することができない。

例をととして、同じ曜日をすべて含む場合、個別に指定する場合、複数指定する場合の書き方を以下に示している。

[3, 2]: 当月の第2水曜日 ($w = 3$)

[6]: 当月すべての土曜日 ($w = 6$)

[5, 2, 4]: 当月の第2第4金曜日 ($w = 5$)

3.2.2 ルールの仕様

祝日の各パターンをルールとして記述する仕様は、表5に示す。各ルールは、日の定義に合わせて、適用期間（年）も記述する。また、ルールは、月別にまとめるので、当月しか適用しない。さらに、同じ祝日もできる限りまとめて簡潔に記述するようにしている。

A. 日の定義。日の定義については、以下の4パターンを用意している。(1) **日付指定**: **day** をキーにして、整数で当月の日付を定義する。(2) **曜日指定**: **dow** をキーにして、曜日指定の仕組みに従い、配列で第何〇曜を定義する。(3) **特別な計算**: **func** をキーにして、計算に使われる関数

表 5 祝日定義ルールの仕様

項目	定義	説明
日の定義	day d	日付 d (整数) を指定する
	dow $[w, n_1, n_2, \dots]$	第 n_1, n_2, \dots の w 曜日を指定する
	days	法律改正等により、適用期間と共に祝日の定義が複数ある場合にこれらをまとめて定義する
	func	特別な計算を行う関数名を指定する。
適用期間	since y	y 年から現在までの各年に適用する
	between $[y_1, y_2]$	y_1 年から y_2 年までの各年に適用する
	including $[y_1, y_2, \dots]$	リストにある年を含む(適用する)
	excluding $[y_1, y_2, \dots]$	リストにある年を除く(適用しない)

を記述する。さらに、(4) 法律改正などにより、同じ祝日の定義は複数存在する場合に対応するため、**days** をキーにして、適用期間と共に複数の定義をまとめることができる。

B. 適用期間。 法律の成立や改正に伴い、ルールの適用期間が変わることがある。適用期間を定義するために、**since**, **between**, **including**, **excluding** を組合せて適用期間をしている。これらの定義をもとに、ある年 y に、あるルールが適用できるかどうかを判断する条件は、式 (1) を使う。

$$y \in Y_{\text{since}} \wedge y \in Y_{\text{between}} \wedge y \in Y_{\text{including}} \wedge y \notin Y_{\text{excluding}} \quad (1)$$

ここで Y_{since} は **since** で指定された年から現在までの年の集合であり、指定がない場合は祝日の法令が最初に成立した年 1948 から現在までの期間とする。 $Y_{\text{between}}, Y_{\text{including}}, Y_{\text{excluding}}$ は、それぞれ **between**, **including**, **excluding** で指定された年の集合とする。 $Y_{\text{between}}, Y_{\text{including}}$ の指定がない場合は、 Y_{since} と同じ集合とする。つまり、 $Y_{\text{including}} = Y_{\text{since}}$ 、または、 $Y_{\text{between}} = Y_{\text{since}}$ とする。 $Y_{\text{excluding}}$ が与えられていない場合は $Y_{\text{excluding}} = \emptyset$ とみなす。

ルールは編集しやすい YAML 形式で記述することとする。YAML (ヤメル、ヤムル) は、人間に理解しやすいように、機械による解釈も可能な形で構造化データを表現できるように標準化された形式である。リスト 1 は YAML 形式で祝日を定義するルールの一部を示している。また、祝日名をカスタマイズしやすいように、祝日名を定義する YAML ファイルを別途用意して、リスト 2 に例示している。

日付指定の祝日として「元日」(1 月 1 日)、曜日指定の祝日として、2000 年以降の「成人の日」(第 2 月曜日)、特別な計算の必要な「春分の日」(3 月)と「秋分の日」(9

```
# ファイル名: holiday_defs.yaml
1: # January
- name: NewYearsDay
  day: 1
- name: ComingAgesDay
  days:
  - day: 15
    between: [1948, 1999]
  - dow: [1, 2] # 第 2 月曜日
    since: 2000
(中略)
8: # August
- name: MountainDay
  days:
  - day: 11
    since: 1948
    excluding: [2020, 2021]
  - day: 10
    including: [2020]
  - day: 8
    including: [2021]
9: # September
- name: RespectForTheAgeDay
  dow: [1, 3] # 第 3 月曜日
- name: AutumnalEquinoxDay
  func: equinox
```

リスト 1: YAML 形式の祝日定義ルール

```
# ファイル名: holiday_names.yaml
Nation: Japan
HOLIDAY_NAMES:
  NewYearsDay: 元日
  ComingOfAgeDay: 成人の日
  NationalFoundationDay: 建国記念の日
  SpringEquinoxDay: 春分の日
(中略)
  SubstituteHoliday: 振替休日
  BridgeHoliday: 国民の休日
```

リスト 2: YAML 形式祝日名定義

月)をそれぞれの項目で定義している。また、1999 までと 2000 以降の「成人の日」はルールが違うので、複数のルールを **days** 項目としてまとめている。このようにルール化することにより、ルールベースの祝日計算が可能になる。法律の改正等に合わせて、ルールを追加・修正するだけで対応でき高い汎用性とメンテナンス性を実現できる。

3.3 祝日の自動計算

これまで述べてきたルールを解析して祝日の日付を決めることができる。既に過ぎた年に既存のルールを適用すれば済むことになるが、ユーザ提供の YAML 形式のルールに基づく計算もできる。

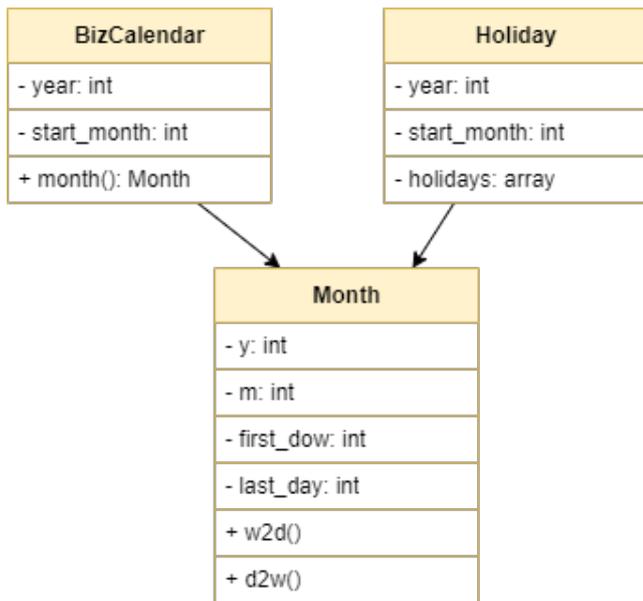


図 2 祝日休日計算のクラス図

3.3.1 ライブラリの構成

ライブラリの基本構成としては、年度を扱うクラス BizCalendar、祝日を計算するクラス Holiday、及び曜日や日付を扱う Month クラスからなっている。BizCalendar には後述する営業日・休業日を扱う機能も組み込まれている。

図 2 は、ライブラリの基本構成を図示している。各クラスに備えた機能/メソッドは表 6 と表 7 に示している。

表 6 Holiday クラスの主要なメソッド

対象	メソッド名	説明
年	parseYear()	対象年に適用可能な祝日定義を抽出し祝日を計算する。月ごとに parseMonth() メソッドを呼び出して祝日を求める
	substitute()	振替休日をもとに祝日リストに追加する
	bridge()	国民の休日を求め祝日リストに追加する
年-月	parseMonth()	一か月分の祝日定義を解析し祝日を計算する。日ごとに parseDay() メソッドを呼び出して祝日を求める
	parseDay()	一日の祝日定義を解析し祝日の日(整数)を求めて祝日リストに追加する。日付指定(day)、曜日指定(dow)、特別な計算(func)に応じて、それぞれのメソッドを呼び出す
	equinox()	春分の日(3月)・秋分の日(9月)を計算する

表 7 Month クラスの主要なメソッド

対象	メソッド名	説明
年-月	w2d(n, k)	今月第 n の k 曜日の日付を計算する
	d2w(n)	今月第 n 日の曜日を計算する
	w2days(dow)	dow 形式 [w, n1, n2, ...] の曜日から日付リストを求める

対象欄は、メソッドの処理対象となるものを意味する。例えば、w2d(n, k) メソッドは、その年と月を対象としているので、その月の何日を返すことで日付を決める。

```

<?php
use KsuCal\Holiday;
use Symfony\Component\Yaml\Yaml;
$holiday = new Holiday(year: 2025, start_month: 4);
$defs = Yaml::parseFile('holiday_defs.yaml');
$holiday->parseYear($defs)->substitute()->bridge();
echo $holiday->isHoliday('5-3')? '祝日': '平日';
print_r($holiday->holidays()); // 祝日を取得
    
```

リスト 3: Holiday による祝日自動計算例

祝日自動計算クラス Holiday の使用例として、リスト 3 に与えている。ルールファイル holiday_defs.yaml を読み込んで解析したあと、substitute() と bridge() を順次呼び出して振替休日、国民の休日を計算し、祝日リスト holidays に追加する。

自動計算に必要な主なメソッドについては、これから説明していく。

3.3.2 w2d(n, k) – 第 n の k 曜日が何日かを求める関数

m 月の第 i 日の曜日を $w_i \in [0, 6]$ で表す整数とする。w1 は 1 日の曜日を表す整数となる。この月の第 n と k $\in [0, 6]$ 曜日は以下の式で求めることができる。

$$w2d(n, k) = \begin{cases} 7n + k - w_1 + 1 & (k < w_1) \\ 7n + k - w_1 - 6 & (k \geq w_1) \end{cases} \quad (2)$$

上記の計算方法を用いて 2025 年 3 月第 3 の木曜日と第 4 の土曜日を求めてみる。表 8 は 2025 年 3 月のカレンダーを表示している。曜日名前の数字は、その曜日を表す整数となる。2025 年 3 月 1 日は土曜日なので $w_1 = 6$ となる。

表 8 2025 年 3 月のカレンダー

日-0	月-1	火-2	水-3	木-4	金-5	土-6
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

第 3 の木曜日を求めるとき、 $n = 3, k = 4$ になる。 $k < w_1$ のため、 $w2d(n, k) = 7n + k - w_1 + 1$ を使と以下のように 20 と正しく求めることができる。

$$w2d(3, 4) = 7 \times 3 + 4 - 6 + 1 = 20$$

また、第 4 の土曜日を求めるとき、 $n = 4, k = 6 \geq w_1$ のため、 $w2d(n, k) = 7n + k - w_1 - 6$ を使うと次のように 22

と正しく求めることができる。

$$w2d(4, 6) = 7 \times 4 + 6 - 6 - 6 = 22$$

しかし、第5の火曜を求める場合には、 $w2d(n, k) = 7n + k - w_1 + 1$ を使って計算すると、次の結果になる

$$w2d(5, 2) = 7 \times 5 + 2 - 6 + 1 = 32$$

当月の最大日数を越えた結果になってしまう。このように、最大日数を越えると、 -1 を返すことにする。

3.3.3 $d2w(n)$ —第 n 日の曜日を求める関数

w_1 は当月1日の曜日を表す整数とする。第 n 日の曜日は以下の式で求めることができる。 $\text{mod } n$ は n で割ったあまりを求める演算である。

$$d2w(n) = (n + w_1 - 1) \text{ mod } 7 \quad (3)$$

表8は2025年3月のカレンダーを使って例示する。2025年3月1日は土曜日で $w_1 = 6$ である。3月13日は $13 + 6 - 1 = 18$, $18 \text{ mod } 7 = 4$ となり、木曜日であることが分かる。また、31の曜日は $(31 + 6 - 1) \text{ mod } 7 = 36 \text{ mod } 7 = 1$ のため、月曜日と正しく計算できた。

3.3.4 春分の日、秋分の日近似計算方法

春分の日(3月の春分日)は、天文学的には「太陽黄経が0度になる瞬間(春分点を通過する日)」を基に決まる。秋分の日、太陽黄経が180度になる瞬間(秋分点を通過する日)に基づいて決められる。日本の春分の日秋分の日、国立天文台が計算し、最終的に政府が決定するが、近似的に求めるための経験則がある[7][8]。以下は1892年-2099年間の近似的な計算方法について説明する。

(1) 「春分の日」の近似計算方法

西暦 Y 年の「春分の日」 D (整数値) は以下の近似式を用いると計算できる。 $\delta(Y) = Y - 2000$ とし、 $[x]$ は床関数(小数部分を切り捨て)とする。

$$D = \left\lfloor 20.8431 + 0.242194 \times \delta(Y) - \left\lfloor \frac{\delta(Y)}{4} \right\rfloor \right\rfloor$$

例えば、2025年の場合($Y = 2025$)、春分の日が3月の何日か、次のように求めることができる。 $\delta(Y) = 2025 - 2000 = 25$, $0.242194 \times 25 = 6.05485$, $\lfloor 25/4 \rfloor = \lfloor 6.25 \rfloor = 6$, $20.8431 + 6.05485 - 6 = 20.89795$, $\lfloor 20.89795 \rfloor = 20$ と求めていくと、結果として、2025年の春分の日は3月20日と計算される。実際の暦でも3月20日であるため、正しく計算できた。

(2) 「秋分の日」の近似計算方法

秋分の日(9月の秋分日)も同様の近似式で計算できる。秋分の日 D (整数値) の近似式は以下の通りである。

$$D = \left\lfloor 23.2488 + 0.242194 \times \delta(Y) - \left\lfloor \frac{\delta(Y)}{4} \right\rfloor \right\rfloor$$

例えば、2025年の秋分の日は9月の何日か以下のように求めることができる。 $0.242194 \times 25 = 6.05485$, $\lfloor 25/4 \rfloor = 6$, $23.2488 + 6.05485 - 6 = 23.30365$, $\lfloor 23.30365 \rfloor = 23$ の順に求めていくと、結果として、2025年の秋分の日は9月23日と計算される。実際の暦でも9月23日であるため、正しく計算できた。

4. 営業日・休業日の自動計算

本節は、営業日・休業日計算の機能について説明する。祝日は、法律などによって決められるものに対して、営業日・休業日は、企業や学校など個別の組織によって決められるものとなる。祝日と同様に、定休日・休業日・営業日もルール化することができ、それに基づいて計算することができる。

4.1 営業日・休業日の指定方法

定休日・休業日・営業日の指定方法は、主に以下の種類がある。(1) **曜日指定**: 曜日で休業日を指定する。例えば、毎月第1、第3月曜日は定休日とする。土日を一律休業日とする。(2) **日付指定**: 日付で休業日を指定する。例: 臨時休業、年末年始特別休業。(3) **特別指定**: 例えば、スポーツなどの施設では国民の祝日を一律営業日とする。

```
2025:
  type: BusinessYear
  start_month: 4 # 2025-4
  default: openday
  offday:
    - name: 休場日
      dow: [2, 2, 4] # 第2第4の火曜日
      since: '4-1'
      excluding: ['8-21', '9-23']
    - name: 定休日
      dow: [5] # すべての金曜日.
      between: ['8-1', '9-30']
    - name: 臨時休業
      including: ['6-13', '11-7'] # 特別な休業日
  openday:
    - name: 営業日
      including: ['10-7', '12-3'] # 例外の営業日
```

リスト 4: 営業日・休業日を定義するルール例

上記のように明示的に指定しない日の対応も必要である。例えば、未指定の日はデフォルトで「営業日」とする。さらに、上記の指定方法で、同じ日に複数指定があった場合に優先順位を決める必要がある。例えば、曜日指定で「休業日」となったとき、日付指定や特別指定で「営業日」と設定された場合がある。このとき、営業日優先の場合は、その日が営業日となる。休業日優先の場合はその日が休業日になる。一般的に、デフォルトと一致する方が優先する。

理由として、指定しなくても営業日になるにも関わらず、わざと指定したのは、特別な理由があると理解できるからである。

営業日・休業日について、年度や優先順位を設定する以外、祝日と同様 YAML 形式でルールを定義できる。リスト 4 は 2025 年度（4 月から始まる）における営業日・休業日を指定するサンプルである。祝日と違うのは、since、between、including、excluding は、日付の指定にも使える。曜日で営業日・休業日を指定する場合のみ、これらの項目は、適用範囲となる。

表 9 BizCalendar クラスの主要なメソッド

メソッド名	説明
<code>isBizDay(d)</code> <code>isOffDay(d)</code>	日付 d は営業日かどうかを返す 日付 d は休業日かどうかを返す
<code>setBizDays(ds)</code> <code>setOffDays(ds)</code>	日付リスト ds を営業日に設定する 日付リスト ds を休業日に設定する
<code>bizDays(s, t)</code> <code>offDays(s, t)</code>	日付 s から日付 t までの営業日を返す 日付 s から日付 t までの休業日を返す
<code>nextBizDay(s, n)</code> <code>nextOffDay(s, n)</code>	起算日 s 後第 n の営業日を求める 起算日 s 後第 n の休業日を求める
<code>setBizPeriod(n, s, t)</code> <code>useBizPeriod(n)</code>	名前 n に日付 s から日付 t までの間を期間として設定する 名前 n の期間を絞込等に利用する

営業日・休業日計算の機能は BizCalendar クラスに組み込まれている。主なメソッドは表 9 にまとめている。bizDays() と offDays() メソッドを使って営業日・休業日を抽出するさいに、引数が与えられていない場合、usePeriod() メソッドで指定した範囲を使って抽出を行う。

使い方例として、リスト 5 に示している。祝日と 7 月の日曜日を休業日にして、期間の定義、指定の期間内の休業日を抽出することができる。

```
<?php
use KsuCal\BizCalendar;
use KsuCal\Holiday;
$holiday = new Holiday(2025, 4);
$cal = new BizCalendar(2025, 4);
//祝日(※1)及び7月の日曜日(※2)をすべて休業日にする
$cal->setOffDays($holiday->holidays());//(※1)
$cal->setOffDays($cal->month(7)->w2days([0]));//(※2)
$cal->addBizPeriod('R7 前期', '2025-4-1', '2025-9-13')
->addBizPeriod('R7 後期', '2025-9-14', '2026-3-31');
$cal->addBizPeriod('R7 上半期', '2025-4-1', '2025-9-30')
->addBizPeriod('R7 下半期', '2025-10-1', '2026-3-31');
$offdays = $cal->useBizPeriod('R7 後期')->offDays();
print_r($offdays);
```

リスト 5: BizCalendar による営業日・休業日計算例

5. 関連研究

Yasumi (日本語で「Holiday」「休み」)は、さまざまな国や州の休日および特別な祝祭日の名称と日付を取得するための簡単な PHP ライブラリである [1]。Yasumi の計算は、国や州ごとに分けられたプロバイダーに基づいており、新しい休日プロバイダーを追加して休日を計算することが容易である。

「JapaneseDate」は、日本の祝日や暦を扱うための PHP ライブラリである [2]。PHP 標準ライブラリ DateTime を拡張した日付時間機能 Carbon に基づいて開発された。

しかし、上記のライブラリは、いずれも既存の法律に基づいたものであり、法律改正による祝日の追加変更や、営業日・休業日の設定等はプログラムの変更や追加の実装が必要となり、汎用性に欠けている。

6. 終わりに

本研究では、日本の祝日および休日を自動計算する汎用的なライブラリを開発した。本ライブラリは、ルールに基づき、各祝日の日付を特定する計算関数、振替休日や国民の休日の自動生成機能、業務要件に応じた柔軟な会計年度設定、さらに営業日や休業日の判定機能を備えている。

今後の課題として、ライブラリの各機能の更なるテストと、その後のリリース等が挙げられる。それにより、他システムとの連携やユーザーが容易に利用可能になりより広い範囲での利用に対応できる。

謝辞

本研究は、小郡市スポーツ振興課よりご支援をいただきました。心より感謝を申し上げます。

参考文献

- [1] Telgenhof, S.: Yasumi: The easy PHP Library for calculating holidays, <https://www.yasumi.dev/> (2024). アクセス日: 2024.12.24.
- [2] Suzunone: JapaneseDate: 日本の祝日、暦など日本での日付処理, <https://github.com/suzunone/JapaneseDate> (2020). アクセス日: 2024.12.20.
- [3] Wikipedia: 年度, <https://en.wikipedia.org/wiki/Year-ending> (2025). アクセス日: 2025.1.10.
- [4] 参議院法制局: 年度について, <https://houseikyoku.sangiin.go.jp/column/column108.htm> (2020). アクセス日: 2024.12.20.
- [5] 所 功: 「国民の祝日」の由来がわかる小事典, PHP 研究所 (2003).
- [6] 金泰虎: 「国民の祝日」と「公休日」を通してみる日韓社会, 博士論文, Konan University (2011).
- [7] Meeus, J. H.: *Astronomical Algorithms*, Willmann-Bell, Incorporated (1991).
- [8] 国立天文台: 理科年表-春分の日・秋分の日, <https://official.rikanenpyo.jp/posts/6496> (2002). アクセス日: 2024.12.20.