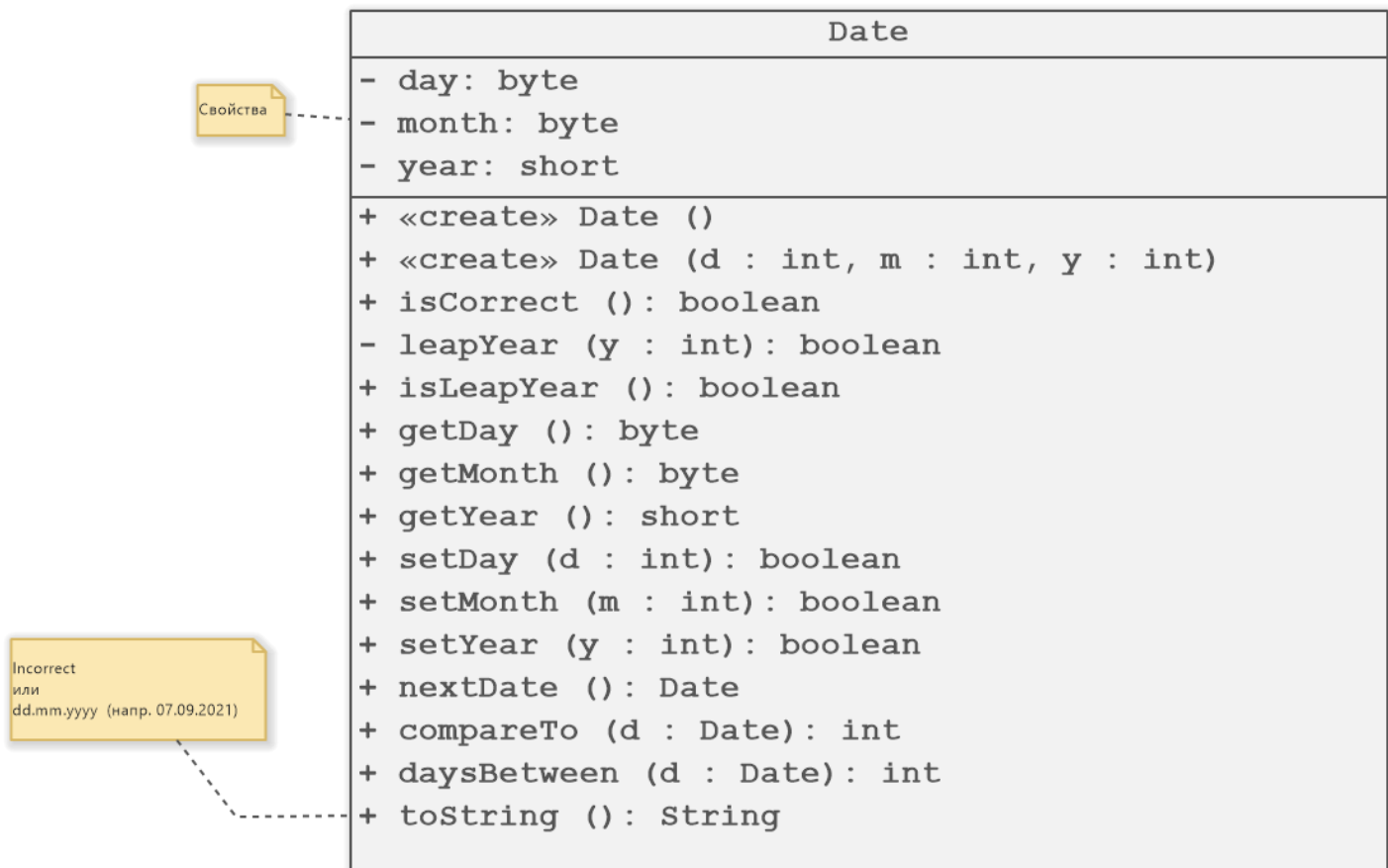


## ПРОЕКТ „ДАТА“



С горната диаграма е описан клас **Date**. Диаграмата има три раздела: име на класа, свойства и методи. Знак „минус“ (–) пред съответния елемент означава, че той е обявен като частен, а знак „плюс“ (+) – че е публичен. Типовете на елементите и на параметрите са записани след името и двоеточие. Думата «create» пред името означава „конструктор“.

Диаграмата описва понятието „календарна дата“. Методът `leapYear(int)` разпознава дали параметърът му е „високосна година“ (ако се дели на 400 или не се дели на 100, но се дели на 4 без остатък). Методите `isCorrect()` и `isLeapYear()` и аксесорите `getDay()`, `getMonth()` и `getYear()` връщат подходяща стойност при некоректна дата. Мутаторите `setDay(int)`, `setMonth(int)` и `setYear(int)` позволяват промяна на свойствата, само ако новополучената дата би била коректна, и връщат информация за „успех“ или „неуспех“ на промяната. Методът `nextDate()` връща обект от клас **Date**, представляващ следващия ден на **this**. Методът `compareTo(Date)` сравнява хронологично **this** с подадения параметър по стандартите за сравняване (връща отрицателно число, ако **this** е по-рано от параметъра, нула, ако двата обекта представляват един и същи ден, иначе връща положително число). Методът `daysBetween(Date)` намира броя дни между **this** и подадения параметър (началото и края на периода се включват). Методът `toString()` връща низ `Incorrect`, ако датата не е коректна, или датата **this**, форматирана по указания в диаграмата начин.

Опишете на Java класа **Date**. Можете (и е редно) да ползвате клас за тестване по време на разработката, но в споделената папка XIA запишете за проверка текстов файл, който съдържа само кода, описващ класа. Името на файла да бъде латинско главно A, следвано непосредствено от номера в класа. Разширението да бъде `.txt`. Примерни имена на файлове за проверка: `A3.txt`, `A27.txt`.

Време за работа: до 90 мин.