

Name:

MatNr:

User:

Hinweis: Denken Sie während der Klausur auch an die **Datensicherung!**

Empfehlung: Arbeiten Sie in kleinen Schritten und sorgen Sie dafür, dass Sie jederzeit ein lauffähiges Anwendungssystem haben.

Allgemeine Vorgaben:

- **Benutzen Sie ausschließlich MS Visual Studio Professional 2013.**
- Verwenden Sie möglichst keine globalen Variablen oder Objekte und nur solche, die als **const** deklariert sind.
- Programmieren Sie C++-standardkonform (z.B. keine Verwendung von **<conio.h>**, kein Aufruf von **system**, keine Ein-/Ausgabe im C-Stil!)
- Verwenden Sie die vorgegebenen Bezeichner; zusätzliche eigene Bezeichner sind 'sprechende Bezeichner'.
- Beachten Sie das 'Principle of least privilege' (z.B. 'const-correctness').
- Die öffentliche Schnittstelle der Klassen darf durch **operator<** und **operator==**-Funktionen erweitert werden.
- Alle Variablen/Objekte werden zur Übersetzungszeit erzeugt und direkt mit ihrem Namen angesprochen, d.h. in Ihrem Programm kommt kein **new**-Operator und keine Zeiger-Variable vor.
- Trennung von Klassendefinitionen und –implementierungen in .h- und .cpp-Dateien.

Zur Bewertung:

Insgesamt können Sie 100 Punkte bekommen.

Die Note 'sehr gut' (1,0 und 1,3) wird für mindestens 93 Punkte vergeben.

Bestanden haben Sie mit mindestens 50 Punkten.

Werden die 'Allgemeinen Vorgaben' nicht eingehalten, gibt es Punktabzug.

Aufgabenstellung:

Sie sollen das "Geburtstagsparadoxon" aus der Wahrscheinlichkeitsrechnung durch eine Simulation überprüfen. Hierbei wird untersucht, wieviele Personen in einer Gruppe sein müssen, damit die Wahrscheinlichkeit dafür, dass es zwei Personen in dieser Gruppe gibt, die am selben Tag Geburtstag haben, größer als 50% ist. Erstaunlicherweise ist dies schon für relativ kleine Gruppen der Fall (daher die Bezeichnung "Geburtstagsparadoxon").

Ihr Simulationsprogramm soll die Klasse **Group** und eine unterstützende Klasse **Date** (s. Klassendiagramme) enthalten:

Group
+ <u>numberOfBirthdays</u> : int - birthdays: Date[numberOfBirthdays] - size: int
+ Group(int) + birthdayOnTheSameDay(): bool + sort(): void + toString(): string

Date
- dayInYear: int - year: int
+ Date() + setRandomDate(): void + getDayInYear(): int + getDayInMonth(): int + getMonth(): int + getYear(): int + toString(): string

Besonderheiten der Methoden sind:

- Der Standardkonstruktor von **Date** soll das erzeugte Objekt auf ein zufälliges Datum setzen, wie unter **setRandomDate** beschrieben.
- **setRandomDate** soll die Datenelemente **dayInYear** und **year** unter den folgenden Randbedingungen setzen: **dayInYear** soll auf zufällige Weise so gesetzt werden, dass jeder der 365 Tage eines Jahres (Schaltjahre werden nicht berücksichtigt) mit der gleichen Wahrscheinlichkeit auftritt. Der Wert für **year** soll zufällig und gleichverteilt aus dem Bereich 1950 bis 2007 (jeweils einschließlich) genommen werden.
- **getDayInMonth** und **getMonth** sollen den im Datenelement **dayInYear** gespeicherten Tag in der üblichen Form 'Tag im Monat' bzw. 'Monat' zurückgeben. Hat **dayInYear** also z.B. den Wert 35, gibt **getDayInMonth** den Wert 4 und **getMonth** den Wert 2 zurück.

- **toString** soll das im **Date**-Objekt enthaltene Datum in der Form ‚04.02.1992‘ als **string** zurückgeben.
- Die (konstante!) Klassenvariable **numberOfBirthdays** hat den Wert **200**.
- Der Konstruktor von **Group** soll das Attribut **size** auf den übernommenen Parameterwert setzen, falls dieser zwischen 1 und **numberOfBirthdays** liegt, andernfalls auf **numberOfBirthdays**. **size** gibt die Anzahl der **Date** –Objekte im Array **birthdays** an, die das aktuelle **Group** –Objekt nutzt (und damit die Größe der aktuell betrachteten Gruppe).
- **birthdayOnTheSameDay** soll **true** zurückgeben, falls **birthdays** mindestens zwei Einträge mit gleichem **dayInYear** enthält, andernfalls **false**. Hierbei sind natürlich nur die ersten **size** Elemente von **birthdays** relevant! (Anmerkung: Das Geburtsjahr ist hier also uninteressant; es geht nur darum, dass zwei Personen am selben Tag Geburtstag haben, egal wie alt sie werden.)
- **sort** soll die relevanten **Date**-Objekte im Array **birthdays** aufsteigend sortieren. Sie können ein Sortierverfahren Ihrer Wahl benutzen; die Sortierung soll zunächst nach **year** erfolgen, bei gleichem Wert von **year** soll nach **dayInYear** sortiert werden.
- **toString** soll den Wert von **size** und die im Array **birthdays** enthaltenen relevanten **Date**-Objekte in einer geeigneten Formatierung als **string** zurückgeben.

Das **Anwendungsprogramm (main)** soll die beiden Klassen testen. Dazu sollen der Reihe nach folgende Aktionen ablaufen (eine Menüsteuerung soll also nicht realisiert werden):

- Erzeugung eines **Group**-Objektes mit 200 Geburtstagen, Sortieren und Ausgabe der sortierten Geburtstage.
- Erzeugung von zwanzig **Group**-Objekten mit jeweils 35 Geburtstagen und Ausgabe für jedes einzelne Objekt, ob mindestens zwei der Geburtstage auf denselben Tag im Jahr fallen oder nicht.
- Erzeugung von jeweils 10 000 **Group**-Objekten beginnend ab der Größe 2 bis zur Größe 50 und Ausgabe, bei wieviel Prozent der 10 000 Objekte mindestens zwei der Geburtstage auf denselben Tag im Jahr fallen. Das Ergebnis soll in Form einer Tabelle ausgegeben werden, deren erste Zeilen etwa so aussehen sollten wie rechts abgebildet. Aus der Tabelle ist dann abzulesen, wie groß ein **Group**-Objekt sein muss (d.h. wieviele Geburtstage es enthalten muss), damit die Wahrscheinlichkeit dafür, dass mindestens zwei auf denselben Tag im Jahr fallen, größer als 50% ist (Geburtsparadoxon).

Gruppengroesse	Mehrfachgeburtstage in %
2	0.23
3	0.67
4	1.80
5	2.52
6	3.88
7	5.86
8	7.38
9	9.53
10	11.49
...	...
50	...

Ergebnis:

Als Ergebnis erwarte ich in jedem Fall ein **ausführbares** Anwendungssystem.

Speichern Sie alle Ihre Projektdateien mit Ausnahme des Debug-Verzeichnisses in Ihrem persönlichen Netzverzeichnis. Das Anwendungssystem muss sich mit diesen Dateien auf "Knopfdruck" erzeugen lassen. Löschen Sie Ihr Projekt am Klausurende **nicht** und fahren Sie den Rechner **nicht** herunter.

Lassen Sie das **Aufgabenblatt** (mit Namen, Matrikel- und Usernummer) an Ihrem Platz liegen.

Bewertung (Punkte):	maximal	erreicht
Klassendefinitionen	12	
getDayInMonth	10	
getMonth	10	
Rest von Date	12	
Konstruktor Group	6	
birthdayOnTheSameDay	14	
sort	12	
toString	6	
main	18	

Gesamtpunkte:

Note: