

Schulklassen

Implementieren Sie die Klassen `Form` (Schulklasse) und `Pupil` (SchülerIn): Ein `Pupil`-Objekt hat die Instanzvariablen `name` (`string`, nicht leer), `elective1` (Wahlfach 1), `elective2` (Wahlfach 2) und `points` (`int`, zwischen 0 und 100 jeweils inklusive). Die Enumeration `Elective` (`Elective::M`, `Elective::E`, `Elective::PH`, `Elective::CH`, `Elective::ART` und `Elective::PHIL`) enthält die möglichen Wahlfächer. Die beiden gewählten Fächer (Wahlfach 1 und Wahlfach 2) müssen unterschiedlich sein. `Elective::ART` und `Elective::PHIL` sind als Werte für das erste Wahlfach (`elective1`) zudem nicht erlaubt.

Für die Klasse `Pupil` sind folgende Methoden und Funktionen zu implementieren:

- Konstruktor(en) mit 2, 3 oder 4 Parametern: Name, Wahlfach 1, Wahlfach 2 und Punktzahl in dieser Reihenfolge. Wahlfach 2 und Punktzahl sind optional mit den Defaultwerten `Elective::PHIL` bzw. 20. Sollte einer der Parameter nicht die Voraussetzungen erfüllen (z. B. Name ist leer, nicht erlaubtes Fach für Wahlfach 1, Wahlfächer nicht unterschiedlich oder Punktzahl nicht im erlaubten Bereich), ist eine Exception vom Typ `runtime_error` zu werfen.
- `int chose(Elective choice) const`: Retourniert 0, 1 oder 2, je nachdem, ob das Wahlfach `choice` gar nicht, als erstes Wahlfach oder als zweites Wahlfach gewählt wurde.
- `int change_points(int amount)`: Addiert den Parameterwert `amount` (der auch negativ sein kann) zu den Punkten des Schülers (`this`-Objekt). Wird dabei der erlaubte Bereich überschritten, so wird bei 100 bzw. 0 abgeschnitten. Retourniert wird in jedem Fall, wie viele Punkte von `amount` durch das eventuelle Abschneiden nicht berücksichtigt werden konnten. Beispiel: Der Schüler habe 10 Punkte und `amount` beträgt -15. Dann wird die Punktzahl des Schülers auf 0 gesetzt und -5 retourniert.
- `void elect(Elective choice, bool first)`: Der Wert der Instanzvariable `elective1` (falls `first true` ist) bzw. `elective2` (falle `first false` ist) des Schülers (`this`-Objekt) wird auf den Parameterwert `choice` gesetzt, so weit sich dadurch keine verbotene Konstellation ergibt. Ist die Auswahl nach den weiter oben vorgegebenen Regeln nicht erlaubt, so ist eine Exception vom Typ `runtime_error` zu werfen und das `this`-Objekt unverändert zu lassen.
- `int get_points() const`: Retourniert die Anzahl der Punkte (`points`) des Schülers (`this`-Objekt).
- `operator<<`: `Pupil`-Objekte müssen in der Form `[name: elective1, elective2, points]` ausgegeben werden, z. B. `[Susan: Mathematics, Philosophy, 70]`. Der vordefinierte Vektor `elective_names` kann für die Ausgabe der Enumerationswerte verwendet werden.

Ein `Form`-Objekt hat die Instanzvariablen `id` (`string`, nicht leer, Bezeichnung) und `pupils` (nicht leere Liste der Schüler, die die Klasse belegen). Für die Klasse `Form` sind folgende Methoden und Funktionen zu implementieren:

- Konstruktor mit zwei Parametern Bezeichnung und Schülerliste. Sollte ein Parameter die Voraussetzungen nicht erfüllen (z. B. Bezeichnung oder Schülerliste leer), ist eine Exception vom Typ `runtime_error` zu werfen.
- `double perc(Elective choice, int minpoints) const`: Retourniert den Anteil in Prozent der Schüler der Klasse (`this`-Objekt), die das Fach `choice` als erstes oder zweites Wahlfach gewählt haben und zumindest `minpoints` Punkte besitzen.
- `operator<<`: Die Ausgabe eines Objekts vom Typ `Form` muss in der Form `[Bezeichnung: {Schülerliste}, total points]` erfolgen. `total points` ist die Gesamtsumme aller Punkte der Schüler in der Klasse. Z. B.: `[5c: {[Susan: Mathematics, Philosophy, 70], [Richard: English, Mathematics, 20], [Mary: Physics, Arts, 30], [Peter: Mathematics, Chemistry, 100]}, 220]`.
- Zusatz für 10 Punkte: Erweitern Sie die Klasse `Form` um die Methode `vector<Elective> popular() const`: Diese retourniert alle Wahlfächer, die von mehr als der Hälfte der Schüler (egal ob als erstes oder zweites Wahlfach) gewählt wurden, in einem Vektor. Die relative Reihenfolge der Einträge im Vektor muss mit der Reihenfolge der Definition der Wahlfächer in der Enumeration `Elective` übereinstimmen.
- Zusatz für 15 Punkte: Erweitern Sie die Klasse `Form` um die Methode `int distribute(Elective choice, int amount)`: Diese verteilt den (nicht negativen) Betrag `amount` von Punkten möglichst vollständig unter den Schülern der Klasse (`this`-Objekt), so dass alle Schüler, die `choice` als zweites Wahlfach haben, denselben Betrag x erhalten und alle Schüler, die `choice` als erstes Wahlfach haben, das doppelte desselben Betrags, also $2x$, erhalten. Der nicht aufteilbare Restbetrag plus die Summe der Punkte, die wegen Überschreitung der 100 Punkte Grenze nicht zugeteilt werden konnten (vgl. Methode `change_points`), ist zu retourneren. Beispiel: Im Zustand, der durch das Beispiel bei `operator<<` gegeben ist, werde `distribute(Elective::M, 103)` aufgerufen. Dann werden jeweils 40 Punkte an Susan und Peter vergeben sowie 20 Punkte an Richard. 3 Punkte verbleiben als Rest. Susan und Peter haben danach jeweils 100 Punkte, Richard 40. Retourniert wird der Wert 53 ($3+10+40$).

Implementieren Sie die Klassen `Form` und `Pupil` mit den notwendigen Konstruktoren, Methoden und Operatoren, sodass jedenfalls das Rahmenprogramm kompiliert und ausgeführt werden kann und die gewünschten Ergebnisse liefert. Achten Sie in Ihren Konstruktoren darauf, dass nur gültige Objekte erstellt werden können. Werfen Sie gegebenenfalls eine Exception vom Typ `runtime_error`.

Für Ihr Programm dürfen Sie **nur** die im vorgegebenen Rahmenprogramm angeführten include-Dateien verwenden!

Instanzvariablen sind `private` zu definieren und die Verwendung globaler Variablen ist (abgesehen von im Rahmenprogramm eventuell bereits definierten) nicht erlaubt! Die Datenkapselung darf nicht durchbrochen werden. Es ist daher unter anderem nicht erlaubt, Referenzen oder Pointer auf private Instanzvariablen einer Klasse nach außen zu vermitteln, `friend`-Deklarationen (mit Ausnahme bei Operatorfunktionen) zu verwenden, oder setter-Methoden zu implementieren, die die Integrität der Daten nicht gewährleisten. Interpretationsspielraum in der Angabe können Sie zu Ihren Gunsten nutzen.

Die Teilaufgaben, bei denen keine Punkteanzahl angegeben ist, gelten als Basisfunktionalität. Für eine positive Beurteilung ist zumindest die Basisfunktionalität zu implementieren. Diese wird mit 30 Punkten bewertet. Die übrigen Teilaufgaben müssen nicht unbedingt implementiert werden, führen aber im Falle einer korrekten Implementierung zu einer entsprechenden Erhöhung der Punkteanzahl.