

Второ контролно по Обектно-ориентирано програмиране

спец. Компютърни науки, 1 и 2 поток, 25.05.2022 г.

Вариант 1

* Липсата на голяма четворка, където тя е необходима, допускането на *memory leak* или *поделяне на памет за даден обект* се наказва с 0 точки на съответната задача. НЕ се допуска използване на `std::string`, `std::vector`.

Задача 1 (1,75). Да се реализира клас за пътешествие с полет със самолет (`AirTravel`). Класът съдържа:

- номер на полет (`flightNumber`) – низ с произволна дължина от тип `char*`;
- времетраенето на полета (`duration`) – задава се в минути;

Да се реализира клас за пътешествие с кола (`CarTravel`). Класът съдържа:

- регистрационен номер на превозно средство (`regNumb`) – представлява масив от четири цифри;
- брой километри, които ще бъдат изминати (`distance`);
- времетраене на пътуването с кола (също с име `duration`) – задава се в минути;

Да се реализира клас за комбинирано пътешествие (`CombinedTrip`), при което има точно едно пътуване с кола и едно пътуване със самолет. Класът притежава всички характеристики на описаните по-горе класове и две допълнителни полета:

- списък от забележителности, които ще бъдат посетени по време на пътуването (`destinations`) – представя се чрез динамичен масив от низове от тип `char*` с произволна дължина.
- свободно време за почивка между пътуванията (`freeTime`) – задава се в минути.

Да се реализира метод `getDuration`, който връща общото време за пътешествието, което включва времето за пътуване с кола, това за пътуване със самолет и свободното време.

Да се реализира метод `print`, който извежда цялата информацията за пътешествието на екрана.

Задача 2 (2,25). Да се реализира система за управление на събития. За всички събития в системата се предполага, че са в един и същ ден (в рамките на 24 часа) и че всички моменти от време се задават с точност до минута.

A) Да се реализират 3 класа, които поддържат следните 3 типа събития:

- **събитие от тип `SimpleEvent`** – събитието се определя от начален и краен час. Максимална продължителност на този тип събития трябва да бъде не повече от 2 часа.
- **събитие от тип `EventWithFixedIntermission`** – събитието се определя от начален час, начало на антракт с фиксирано време от 20 минути и краен час. Максималната продължителността на събитието (без времето за антракта) е не повече от 4 часа.
- **събитие от тип `EventWithIntermission`** – събитието се определя от начален час, начало и край на антракта и край на събитието. Продължителността на събитието (без времето за антракта) трябва да е не повече от 6 часа, а антрактът трябва да е с продължителност между 30 минути и 1 час.

Представянето на часовете за събитията е по ваш избор. Ако се подадат неправилни данни при създаването на съответните събития, програмата да извежда съобщение за грешка и да инициализира събитието със стойности по подразбиране.

Б) Да се реализира контейнер `EventManager`, в който се пази колекция от произволен тип събития с фиксиран максимален капацитет, т.е. контейнерът не може да се преоразмерява след първоначалното си създаване. Да се дефинира операция добавяне на събитие в контейнера със следната сигнатура:

`bool addEvent (<подходящ_тип>* event)`. Добавянето се извършва чрез записване на указателя в контейнера (т.е. контейнерът “открадва” паметта, заделена за събитието) и само ако има достатъчно място.

Бонус: Да се реализира клониране на събитието в контейнера вместо паметта да се “краде”.

В) Да се реализира метод `ongoingEvents`, който по подаден час намира броя на всички събития, които се провеждат в дадения час. Ако едно събитие е в антракт, то не се провежда в настоящия момент.

Второ контролно по Обектно-ориентирано програмиране

спец. Компютърни науки, 1 и 2 поток, 25.05.2022 г.

Вариант 2

* Липсата на голяма четворка, където тя е необходима, допускането на memory leak или разделяне на памет за даден обект се наказва с 0 точки на съответната задача. НЕ се допуска използване на std::string, std::vector.

Задача 1 (1,75). Да се реализира клас за пътешествие с влак (TrainTravel). Класът съдържа:

- номер на линия (trainLine) – низ с произволна дължина от тип char *;
- цена на пътешествието с влак (price) – дробно число с точност две цифри след десетичната точка;

Да се реализира клас за пътешествие с кола (CarTravel). Класът съдържа:

- регистрационен номер на превозно средство (regNumb) – представлява масив от пет цифри;
- брой километри, които ще бъдат изминати (distance);
- цена на пътешествието с кола (също с име price) – дробно число с точност две цифри след десетичната точка.

Да се реализира клас за комбинирано пътешествие (CombinedTrip), в което има точно едно пътуване с влак и едно пътуване с кола. Класът притежава всички характеристики на описаните по-горе класове и две допълнителни полета:

- списък от забележителности, които ще бъдат посетени по време на пътуването (destinations) – представя се чрез динамичен масив от низове от тип char* с произволна дължина;
- пари в брой за свободното време между двете пътешествия (cash) – дробно число с точност две цифри след десетичната точка.

Да се реализира метод getPrice, който връща общата цена на пътешествието, която включва цената за пътуване с влак, тази за пътуване с кола и парите в брой, които трябва да носим в нас.

Да се реализира метод print, който извежда цялата информация за екскурзията на екрана.

Задача 2 (2.25). Да се реализира система за управление на събития. За всички събития в системата се предполага, че са моменти (с пренебрежима продължителност), случват се в един и същ ден (в рамките на 24 часа) и че всички моменти от време се задават с точност до минута.

А) Да се реализират 3 класа, които поддържат следните 3 типа събития:

- **събитие от тип SingleEvent** – събитието се определя от точен час, в който се е случило.
- **събитие от тип HourlyEvent** – събитието се определя от начален час и брой повторения на всеки 60 минути от началния час. Събитието може да се повтори не повече от 5 пъти.
- **събитие от тип RecurringEvent** – събитието се определя от начален час, период на повторение и брой повторения. Периодът на повторение трябва да е между 10 минути и 3 часа, а броят повторения да е не повече от 5.

Представянето на часовете за събитията е по ваш избор. Ако се подадат неправилни данни при създаването на съответните събития, програмата да извежда съобщение за грешка и да инициализира събитието със стойности по подразбиране.

Б) Да се реализира контейнер EventManager, в който се пази колекция от произволен тип събития с фиксиран максимален капацитет, т.е. контейнерът не може да се преоразмерява след първоначалното си създаване. Да се дефинира операция добавяне на събитие в контейнера със следната сигнатура:

bool addEvent (<подходящ_тип> event). Добавянето се извършва чрез записване на указателя в контейнера (т.е. контейнерът “открадва” паметта, заделена за събитието) и само ако има достатъчно място.

Бонус: Да се реализира клониране на събитието в контейнера вместо паметта да се “краде”.

В) Да се реализира метод occurredEvents, който по подаден начален и краен час намира броя на всички събития, които са се случили в дадения интервал.