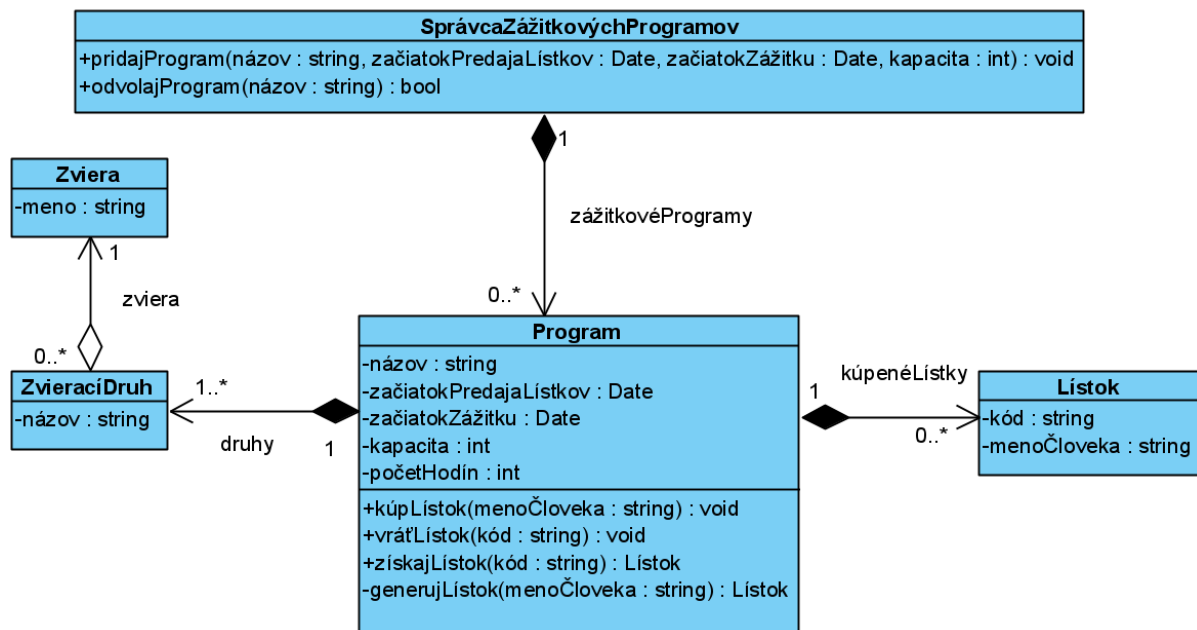


Uvažujte systém, ktorý slúži na správu zážitkových programov v ZOO. Tento systém umožňuje návštevníkom kúpiť si lístky na daný program, taktiež ale i kúpené lístky vrátiť. Každý program je v danom okamžiku buď aktívny (je možné si na neho kúpiť lístky, aj ich vrátiť) alebo neaktívny (s lístkami sa už manipulovať nedá). V aktívnom stave je zážitkový program teda od zahájenia predaja lístkov do začiatku programu. Pokiaľ nastane situácia, že v deň zahájenia programu nebude mať žiaden návštevník kúpený lístok, program sa automaticky zruší. Pokiaľ niekedy po zahájení predaja lístkov 50% sľúbených zvierat ochorie, ZOO je povinná tento zážitkový program zrušiť (počas priebehu programu sa nekontroluje, či zvieratá ochoreli). Po odvolaní alebo po ukončení zážitkového programu sa program zmaže. Pokiaľ sa danému programu naplní kapacita, lístky už nie je možné nakupovať, iba ich vracieť.

V UML notácii vytvorte stavový diagram modelujúci vyššie popísaný životný cyklus objektu **Program**, ktorý korešponduje zadanému diagramu tried. Stavový diagram modelujte do takej úrovni, ako to tento diagram tried dovoľuje. Diagram tried neupravujte. Popis asociácie v ňom predstavuje názov atribútu v referencii k danej triede, takže napríklad **Program** má v sebe kolekciu *druhy*.



Poznámka: na kolekciách môžete volať všetky základné metódy (pridaj(), získaj(), uprav(), zmaž(), počet() a pod.).

Tip: na písanie dlhších akcií, ktoré sa opakujú vo viacerých prechodoch môžete využiť Note, kde definujete, čím daný reťazec akcií substituujete (pre lepšiu čitateľnosť, a aby sa vám váš diagram vošiel na papier).

Príklad použitia:

