# Домашна работа №2

В тази задача трябва да реализирате система за потребителски интерфейс.

Интерфейсът се състои от форми (клас Form). Всяка форма има име и съдържа нула или повече контроли (клас Control). Контролите могат да бъдат за въвеждане (InputControl) или за извеждане на информация (OutputControl). Входните контроли могат да получат информация от потребителя, която да промени тяхното състояние. Изходните контроли могат да получат информация само от програмата. За всяка контрола може да се генерира описание. Възможно е да има и комбинирани контроли (такива, които могат да въвеждат и да извеждат информация, IOControl).

Всички контроли имат:

* уникален идентификатор – цяло число. Трябва да гарантирате, че в никой момент няма две контроли с еднакъв идентификатор;
* относителна позиция в контейнера, в който са разположени – двойка цели числа. Мислете си за позициите и размерите като за клетки в текстов интерфейс – една клетка е един символ. Координатната система започва от горния ляв ъгъл, който е с координати (0, 0), и нараства в посока на четене – надясно и надолу.
* размер, който се определя според типа на контролата;
* текст, който има различна роля в различните контроли.

Контроли са:

* Label – само за извеждане на информация. Показва един ред текст. Размерът е единица на височина, а дължина – колкото е дължината на текста в контролата. Текстът е описанието на контролата. Той се използва и при визуализация.
* CheckBox – контрола за въвеждане на информация. Има свойство дали е включен или не. Размерът е единица на височина и толкова на дължина, колкото е текстът, зададен за нея, плюс четири символа – за маркер за избор и един интервал преди текста. Описанието на контролата включва текста, зададен при нейното създаване. Състоянието на контролата е изборът на потребителя („Checked“ или „Unchecked“).
* RadioButton – контрола за въвеждане. Текстът, зададен за контролата, съдържа няколко опции. Контролата позволява избор на точно една от тези опции и има свойство коя от опциите (или никоя) е избрана в момента. Размерът е толкова реда на височина, колкото са опциите, и ширина – колкото най-дългата опция, плюс още четири символа. Състоянието на контролата се определя от избраната в момента опция. Описанието на контролата включва всички зададени опции.
* TextBox – контрола за въвеждане и извеждане на текст. Размерът се подава при създаването ѝ. При извеждане на текст в нея, ако той е по-дълъг от указаната ширина, да се изведе само първата част, която се събира в нея. Ако текстът е с много редове, но контролата няма достатъчна височина, се извеждат само първите редове. Описанието на контролата е целият текст, който се съдържа в контролата. Състоянието на контролата е текстът, който може да бъде визуализиран в нея.
* ListBox - контрола за въвеждане. Текстът, зададен за контролата, съдържа множество предварително дефинирани елементи. Контролата позволява да се избере точно един от тях. Има размер един ред на височина и ширина, която зависи от дължината на избрания елемент. Описанието на контролата включва всички предварително дефинирани елементи. Състоянието на контролата се определя от избрания елемент. По подразбиране е избран първият елемент. При визуализация се показва текстът на избрания елемент.
* Panel – контрола за въвеждане и извеждане. Панелът е колекция от произволни контроли, всяка с позиция в рамките на панела. Размерът на панела е минималният, който може да покрие всички, разположени в него контроли. В началото панелът е празен, но контролите могат да бъдат добавяни последователно към него. Текстът, който се подава при неговото създаване, служи като заглавие. Описанието на панела включва неговото заглавието. Състоянието на панела е конкатенация (обединение) от идентификаторите на всички входни контроли и техните състояния, разделени със символа ‚|‘. Информация за изходна контрола, част от панела, може да бъде зададена чрез низ, съдържащ идентификатор на контролата и самата информация, които да бъде записана.

Вашата реализация трябва да позволява да се създава форма и в нея да се добавят множество контроли. Не е нужно това да става с потребителски интерфейс. Формите трябва да бъдат конструирани програмно, например от текстово описание, и това да е показано с конкретен пример в кода ви (тестова функция).

Също трябва да можете да извеждате форма на екрана (в текстов вид). Извеждането е според позицията на контролите в нея, започвайки от горния ляв ъгъл с координати (0, 0).

Например, ако имаме форма, която се състои от:

1. етикет с координати (0, 0) и текст „Input:”
2. текстово поле с координати (0, 8), размер (1, 10) и текст „Test“
3. радио бутони с координати (1, 0) и опции „Yes”, “No”, “Maybe”, за който е избрана опцията “No”,

тя може да се изведе по следния начин:

Input: [Test ]

[ ] Yes

[X] No

[ ] Maybe

Класът Форма трябва да има оператор за индексиране [], който по идентификатор на контрола връща подходящ указател.

Трябва да имате метод, чрез който да въведете от потребителя входни данни за всяка входна контрола от формата.

За формата трябва да можете да извлечете като текст информацията от всички нейни входни контроли. Този текст трябва да е съставен от идентификатор на всяка контрола и въведената за нея информация от потребителя (състоянието на контролата). Ако това е текстово поле, това е съдържанието му. Ако е радио бутон, това е текстът на избраната опция. Ако е лист, това е текстът на избраната опция, а ако е чек контрола – текст “Checked” или “Not Checked”. За панелите това е конкатенация от текстовете на всички входни за тях контроли, разделени със символ ‘|’.

По подаден идентификатор на контрола и текст, трябва да може да се промени текстът на съответната изходна контрола.

Всяка входна контрола трябва да реализира възможност за въвеждане на данни от потребителя по подходящ начин.

Подсигурете коректно поведение на всички ваши класове, независимо дали подадените им данни са коректни или не.